

# An Artificially Intelligent Path Planner for an Autonomous Robot

Navya Prakash <sup>1</sup>, Gerard Deepak <sup>2</sup>P.G. Student, Department of Studies in Computer Science, University of Mysore, Manasagangotri, Mysore, India<sup>1</sup>P.G. Student, Department of Computer Science and Engineering, UVCE, Bangalore University, Bangalore, India<sup>2</sup>

**ABSTRACT:** This work focuses on detection of an obstacle in the path of a robot. It uses a wireless camera with a receptor and a red laser light with high intense. It stores the path planning information in the form of a video. It provides the distance of the obstacle using the differentiation of the foreground, background of the image. It provides the depth estimation map for the image frame with the presence of an obstacle. It demonstrates the shortest path from a source to the destination using A star algorithm simulation.

**KEYWORDS:** Path Planning, Video store, Distance of an obstacle, Depth estimation, Shortest path, A star algorithm.

---

## I. INTRODUCTION

Robotics is the field of study of the design, construction and the way of use of robots for a particular task. The robots can operate equipment to much higher precision than humans. As humans can plan their path in any environment, the robots are to be given an artificial intelligence to plan their path in any environment. To achieve this, an autonomous robot has to be used to detect and avoid obstacles, plan its path in the shortest distance. There are many existing systems to detect obstacles using sensors. To get more accurate results, usage of an efficient system is necessary. The work presented will be able to achieve the accurate results. The detection of an obstacle will be guided by a wireless camera with a receptor and a red laser. The distance of the obstacles from the current position of the robot will be calculated in order to plan the shortest path for the robot to navigate. These autonomous robots find varied applications such as in the field of agriculture to detect pests, in military applications to detect suspicious objects. The path planning information has to be stored in order to tutor another autonomous robot.

Obstacle detection is the key feature by which a robot is characterized. It's one of the key functionalities which must be constituted in the behaviour of the robot. The detection is mostly generalized and the exact distance metric calculation for the distant obstacle is totally absent in the existing models. The non-availability of dynamic video recording mechanism for the entire track of robot navigation is a mandate to be incorporated into an efficient robot. Hence, the exact measure of obstacle distance calculation and computation of essential robot features for achieving dynamism in the robot is the key implementation in this paper.

The key challenge of overcoming the distance metric calculation of the obstacle is achieved by using a red laser light and finding out the object distance based on the intensity factor of the red laser light. A special algorithm is implemented to compute the shortest path dynamically from the obstacle and tracking of the robot's pathway is also implemented.

Paper is organised as the section II describes about the algorithms used for implementing this paper. The section III describes the results obtained after conducting the experiment.

## II. RELATED WORK

Latha D U et al.<sup>1</sup> explains a method to detect obstacles using a single camera and a laser. The movement of the robot depends on the position of the obstacle in the image frame. It uses serial communication to communicate between the robot and base station. The image frame is divided into 6 regions for identification of obstacle region, it acts as a depending factor for the rotation and the movement of the robot.

Deepu R et al.<sup>2</sup> proposed a method of detecting an obstacle using a single camera with a laser and generate a map for the navigation of the robot. It explains about the working of the floor segmentation from an image. The absence of the laser light on the floor is made as a hole and is skipped.

## III. PROPOSED WORK

The Fig 1 shows the sequence diagram of the proposed work. In this work, it explains the input, processing and the output obtained. A video stream is taken as an input, in the video the red laser dot fallen on the obstacle is also included. All the image frames are extracted from the video stream and are processed for the detection of the red laser dot on an obstacle. If an obstacle is found, then it would process further to find the region of the obstacle and it would decide the movement of the robot<sup>1,2</sup>. Among all the image frames, one image frame is selected to find the depth estimation map. The video will be accessed again, to find the distance of the obstacle. The entire video is stored as it forms the information of the path planning by the robot.

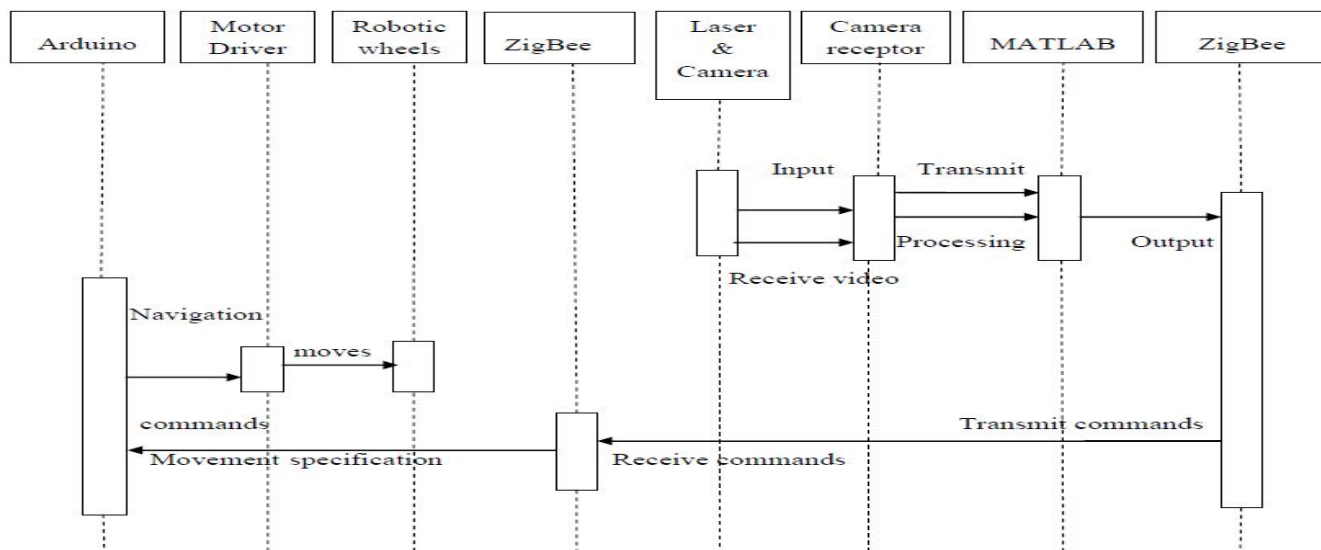


Fig 1: Sequence diagram of the work

The fig 1 represents the sequence diagram of the execution of the entire system. The flow, input and the output of the execution procedure is shown in the figure.

### 3.1 Video Store Algorithm

- Create an object of video input type.
- Start the execution of the object.
- Record the video stream.
- Store or record the video in "avi" format.
- Save the video in the local disk of the base station.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

The video object is started when the robot's execution begins and stopped till the robot stops its execution. The entire navigation of the robot by autonomously planning its path in an environment is stored and made available to watch either live or in recorded form and stored in the disk of the system.

### 3.2 Distance of an obstacle

- Start the wireless camera and view the video.
- Calibrate the wireless camera.
- Construction of a rectangle is made to identify the region of interest to find the distance in each image frame captured at a given time interval from the video.
- Differentiate the foreground from the background in the captured image frames using the rectangle construction in the foreground.
- Find the area of the rectangle for every image frame. The area forms the approximate value of the distance of the obstacle from the robot.
- The distance of the wall from the robot is also taken care of, as the red laser dot falls on the wall; it differentiates the wall from the red laser dot. Only if the red laser dot falls on the wall, distance is calculated otherwise the distance is 0.

The distance of an obstacle will be calculated from the position of the robot using a single wireless camera. The camera has been calibrated; this is done with the image analysis of the captured input image. The foreground and the background identification of the image is a challenging task and were performed with ease with the help of built-in functions Image Processing toolbox.

The robot identifies the walls as an obstacle as the foreground and the background of the image of the wall will be similar. The robot encounters the wall and it takes 180° turn either right side or left side, the decision is provided by the human to the robot. The distance of the wall will be noted as 0, as the robot identifies the wall being near that at a very short distance. In order to avoid the collision of the robot and the wall or stairs; they are treated and identified as the obstacles. The stairs are detected and avoided much before the robot tries to reach the stairs. In the foreground of the image with the stairs, there would be many continuous stairs but only the first stair encountered by the robot in its video will be taken into consideration.

### 3.3 Depth estimation map

- Create an object of the type to represent the image snapshot.
- Use Gaussian filters to find the derivatives of the pixel values.
- Create the 1<sup>st</sup> derivative of the Gaussian filters and set the critical value.
- Create the 2<sup>nd</sup> derivative of the Gaussian filters and set the critical value.
- Create the 3<sup>rd</sup> derivative of the Gaussian filters and set the critical value.
- Plot all the derivatives of the Gaussian filters in-order to get the information of the depth estimated value of the snapshot to confirm the existence of an obstacle in the navigational path.

The 3D depth map estimation was calculated using the Gaussian filters with the help of functions in the Image Processing toolbox. Gaussian filters are filters whose response is a Gaussian function. They are used as an operator in Image Processing for smoothing an image by 'blur' and remove detail and noise. The derivatives of the Gaussian filters are calculated for each and every pixel in the input image. The sigma value i.e. the standard deviation of the Gaussian distribution value is changed for few instances and all the results are stored. The map is constructed in a 3-Dimensional representation.

The one-dimensional Gaussian filter has an impulse response given by

$$g(x) = \sqrt{a/\pi} \cdot e^{-a \cdot x^2} \quad (1)$$

In two dimensions, it is the product of two such Gaussians, one per direction:

$$g(x,y) = 1/(2\pi\sigma^2) \cdot e^{-x^2+y^2/2\sigma^2} \quad (2)$$

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 10, October 2015

The 'x' is the distance from the origin in the horizontal axis, 'y' is the distance from the origin in the vertical axis and ' $\sigma$ ' is the standard deviation of the Gaussian distribution.

### 3.4 Shortest path using A star algorithm

- Create a graph of the type *grid* to plot the graph.
- Select the *initial position* of the vehicle using the pointer of the mouse of the base station.
- Select the *final position* of the vehicle using the pointer of the mouse of the base station.
- Select the *obstacle positions* for the vehicle using the pointer of the mouse of the base station.
- Calculate the pixel difference between the initial point and to the first occurrence of the obstacle and between each and every obstacles present in the grid of the graph plot and for the final obstacle position to the target position.
- If the *difference calculated is less* when on a comparison with other obstacles, then there *exists a shortest path* from one point to another else discard that value and calculate the difference between other point on the grid of the graph.

A star algorithm is a computer algorithm that is widely used in path finding and graph traversal, the process of plotting an efficiently traversable path between multiple points, called nodes. Noted for its performance and accuracy, it has widespread use. However, in practical travel-routing systems, it is generally outperformed by algorithms which can pre-process the graph to attain better performance, although other work has found A\* to be superior to other approaches. It is an extension of Dijkstra's algorithm. A\* achieves better performance by using heuristics to guide its search.

The algorithm for the shortest path simulation from a start node to a destination node in the presence of one or many obstacles is executed. Every obstacle distance is calculated from the start node to the destination node. The nearest path information is stored and its simulation is shown as a real-time interaction of human and system. The implementation of the shortest path algorithm is made on the robot having defined its start point to be the point when the robot starts its execution and the end point is identified as the point in the path when they are no more obstacles encountered during the robot's navigation. Human intervention is much needed to help the robot to work efficiently to find the shortest path in the real-time. The simulation is studied by the human and followed the same for the current environment of the robot's path planning. The navigation of the robot is duly controlled and commanded by the human.

The robot might encounter a wall during its shortest path planning and then it turns  $180^{\circ}$  either right or left side as commanded by the human. It starts its navigation again as the wall to be barrier to navigate further and sometimes it marks the end of the path in an environment; at this situation the robots also stops its execution.

## IV. EXPERIMENT AND RESULTS



Fig 2: Processed image after detection of the obstacle

The Fig 2 represents the processed image of the detection of an obstacle even though the obstacle colour is red. This would prove the Path Planner accuracy level of working.

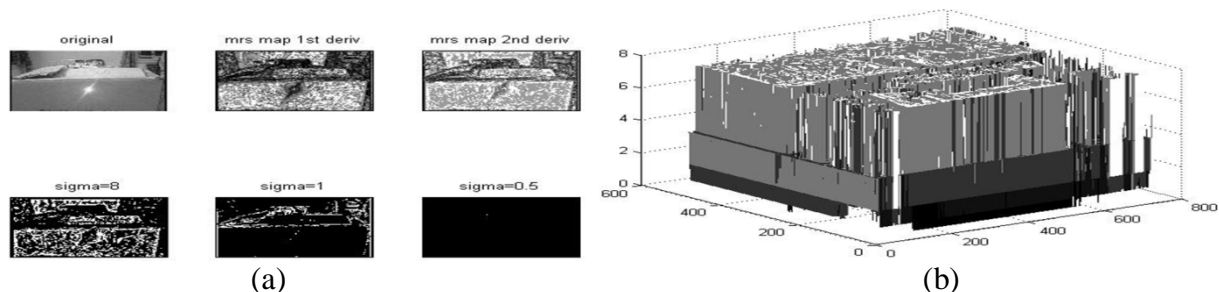


Fig 3: (a) Gaussian filtering results  
(b) 3D Depth estimation map

The Fig 3(a), 3(b) represents the out of the depth map of an obstacle detected by the robot. The various sigma values are considered for the Gaussian Filters to get more smoothing of the input image.

## V. CONCLUSION

This paper provides methods to detect and avoid obstacle for an autonomous robot and plan its path of navigation in an environment. The video is stored from the beginning of the execution till the end; it is preserved to extract information for the path travelled by the robot. The distance of the obstacle calculated is noted to be approximate. The depth estimation map is also noted to be approximated. The shortest paths from the current position of the robot till there are no more obstacles to be detected were found to be accurate.

## REFERENCES

- [1] Latha D U, Deepu R, Honnaraju B, Path planning for a robot with single camera, International Journal For Technological Research In Engineering Volume 1, Issue 11, July-2014.
- [2] Deepu R, Honnaraju B, Murali S, Path Generation for Robot Navigation using a Single Camera, International Conference on Information and Communication Technologies (ICICT 2014), Elsevier, Procedia Computer Science.
- [3] Apoorva Joglekar, Devika Joshi, Depth Estimation Using Monocular Camera, International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1758-1763.
- [4] Heramb Nandkishor Joshi, Prof J. P. Shinde, An Image Based Path Planning Using A – Star Algorithm, International Journal of Emerging Research in Management & Technology ISSN: 2278-9359.
- [5] Shahed Shojaeipoura, Sallehuddin Mohamed Harisa, Robot Path Obstacle Locator using Webcam and Laser Emitter, Elsevier, Physics Procedia 5 (2010) 187–192.
- [6] Punith Kumar M B, Dr. P.S. Puttaswamy, Video to frame conversion of tv news video by using matlab, International Journal of Advance Research In Science And Engineering, IJARSE, Vol. No.3, Issue No.3, March 2014.
- [7] O Hachour, Path planning of autonomous Mobile robot, 2008.
- [8] G. Zhang S. Ferrari M. Qian, An Information Roadmap Method for Robotic Sensor Path Planning, 2009.
- [9] Abhishek Chandak, Ketki Gosavi, Shalaka Giri, Sumeet Agrawal, Mrs. Pooja Kulkarni, Path Planning for Mobile Robot Navigation using Image Processing, June 2013.
- [10] Iv'an Mac'ial, Generalized Computation of Gaussian Derivatives Using ITK, Release 1.00, October 5, 2007.
- [11] Arijit Mallik, Subhendu Das, Design and development of an acoustic distance finder in Android SDK and MATLAB, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 1, January – February 2013 ISSN 2278-6856.
- [12] Pallav Garg, Suresh Yerva, Krishnan Kutty, Relative Depth Estimation using a Rotating Camera System.