PATH NAVIGATION BY DRAWING PATH ON MAP IN STATIC ENVIRONMENT

Dr. E. Logashanmugam[1], K. Joseph Pranadeer Reddy[2]

1Professor at Dept. of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, INDIA
2 Student at Dept. of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, INDIA

ABSTRACT**—** Path planning plays a prominent role in the field of autonomous mobile robots to find an optimal path between start and destination by either plotting the shortest path or minimizing the amount of turning or minimize the amount of braking, depending on the application. There already exist several Path planning algorithms each of which draws a significant amount of computational power. So, we have come up with an approach to reduce the number of iterations performed by the processor to find the path. In this approach, the desired path to be followed by the mobile robot is in control of the user. This input by the user is given by drawing the path on the generated map and using this information, path navigation is carried out. An Android application has been designed so that the map can be generated using any smartphone.
Keywords— Path Planning, Color Detection, Raspberry pi, Autonomous robot navigation.

I. Introduction

In mobile robotics, path planning is considered as an important task. Its wide usage can be found in Self-driving cars, google maps, etc. It is also being used for indoor services in industries, hospitals, and homes. The main requirement of optimal path planning is the accessibility of environmental information. Only after obtaining the map of the environment, Path planning can be employed. The planned path has to be collision-free while maintaining the accuracy in localization.

II. existing system

For path planning, we somehow need to represent the environment on a map with marked obstacle space which can be analyzed by the robot. This space of the environment can be generated by creating a virtual occupancy grid and plotting the obstacles on it. The sensor systems also play a crucial role by scanning the environment to generate this map [1] [2]. Therefore, this method of generating data consumes time and effort. Recently, the use of lidar to generate space has become popular which comes with added expense, complexity, and accuracy [1].

Many Algorithms have been proposed and developed to achieve a better-planned path for a mobile robot. These algorithms include Dijkstra's Algorithm, Visibility Graphs, Cell Decomposition Technique, A* and Modified A* Algorithms, etc.

The path predicted by these algorithms may not be a desired path by the user. Even though, the plotted path is shortest or reduces the power consumption, the freedom to drive the mobile robot in the desired path is lost.

III. Overview of the proposed system

A. Proposal

The sole purpose of this proposal is to increase the comfort generating the map for the user and also, increase the flexibility to drive the robot in the desired path. The same planned path can be used by the robot to return to its starting point.

An app has been designed to generate maps by capturing pictures of the environment and the desired path of travel is drawn on this map by the user, using which path planning is done.

B. Novelty

- Using the APP to generate a map, to make the process user-friendly.
- Driving the mobile robot in the desired path with minimal effort.
- Less time consumption for planning the path.
- Reduction of the number of components to analyze the environment and generate an occupancy grid.

IV. Material Used

The different components, apps, and communication mediums used to demonstrate the proposed Path Planning Technique has been delineated below.

A. APP

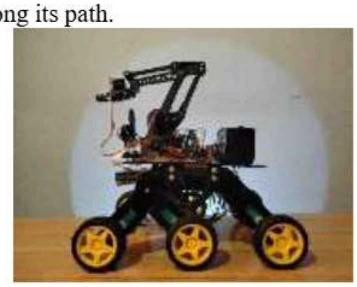
Using MIT APP Inventor, an android application has been designed to communicate information to the processor. The app has been designed to provide commands to two raspberry pi processors using the webserver. Also, the required occupied space on the generated map is done in the app by plotting the start, end, and line of travel. The presence of obstacles can also be marked on the grid map.

B. COMMUNICATION OF GENERATED MAP

The communication means between App and Raspberry pi is achieved using the webserver. The long-range, steady communication strength and simultaneous control of multiple devices using webserver provide far more advantages compared to using Bluetooth Communication. Using different port IDs, multiple devices can be operated.

C. ROVER- ROCKER BOGIE MECHANISM

For demonstration, a mobile robot that has a Rocker-bogie mechanism interfaced with a robotic arm has been used. The Rocker-bogie mechanism aids movement across various surfaces and the Robotic arm assists in the manipulation of objects along its path.



D. RASPBERRY PI- MICROPROCESSOR

CAMERA: Depending on the user's comfort, either one of the two cameras can be used. First is a Pi camera interfaced with raspberry pi can be used to capture the environment. This Pi is placed at a fixed position and height, in an environment. Alternatively, the camera provided in a smartphone can be used to perform the same function. The Camera is only used for obtaining a picture of the environment.

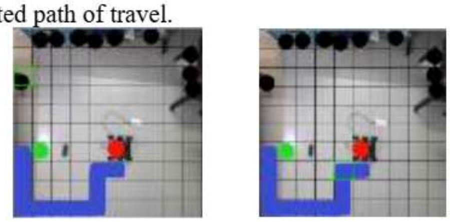
RASPBERRY PI- MICROPROCESSOR: The Raspberry Pi controls the entire operation. The webserver is initiated by this processor. One Raspberry Pi is fixed on the ceiling of the room which is used only for taking pictures of the environment when commanded by the user using the app. Another Raspberry pi is installed on the rover which is used for performing path planning.

V. Techniques Used

Different algorithms and techniques used in this proposal have been elaborated below.

A. SLIDING WINDOW

A Sliding window which has a rectangular region with a fixed width and height plays a crucial role in image classification. This window is slide across an image, allowing the user to localize accurately and analyze that part of an image. The necessary technique is applied across this window to determine any element of interest. Here, this sliding window is combined with Dominant Color detection to localize and detect the plotted path of travel.



B. DOMINANT COLOR

Dominant Color returns the part of color which is mostly occupied in a region. K-means clustering is used to obtain the dominant color value. With the right value of *k*, the centroid of the largest cluster will return a pretty good representation of the region's dominant color.

The 'Average Color' is the sum of all pixels divided by the number of pixels. Using 'Average Color' instead of 'Dominant Color' would yield a color different from the most prominent visual color. Hence, 'Dominant Color' is employed, which results in the RGB values of the color.

This array of values is passed through the bounding values of HSV to obtain the name of the color. Using this, the start, end, and line of travel are tracked. In this demo, if the dominant color is 'Red', then it is marked as a START point. If the dominant color is 'Green, then it is marked as an END point. If the dominant color is 'Blue, then it is marked as TRAVEL point.

Different lists are created to store the coordinate values of the required color.

C. Path Planning

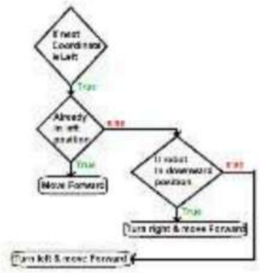
After acquiring the three lists- START, END, and TRAVEL, which contain its respective coordinates, these coordinates are assembled in an order to form a path of travel.

Firstly, the coordinate from the 'START list is taken and a neighboring coordinate is checked for, in the 'TRAVEL' list. After acquiring the neighboring coordinate. Then again, a neighboring coordinate to the present one is checked in the 'TRAVEL' list until the entire list runs out of elements. These are stored systematically in an order which forms the planned path.

```
Planned Path:
[(6,7), (6,8), (5,8), (5,9), (5,10), (4,10), (3,10), (2,10), (1,10), (1,9), (1,8), (1,7), (2,7)]
```

D. *Localization*

After obtaining the path, based on the next coordinate's position with respect to the present one, either up, down, left or right is indicated. Based on this, the rover travels physically on the surface. Different cases have been taken into consideration for accurate outputs. A Flow Chart of such a case has been shown below.



VI. Employment of Technique and Working Mechanism

Firstly, to generate the map of the environment, a photo of the landscape is taken in a birds-eye view (Top-down visual), using the App developed in MIT app inventor. In such a scenario, the position, and orientation of the camera while capturing play a key role. To avoid such a hassle, a raspberry pi interfaced with Pi camera is fixed on the ceiling, positioned, and oriented to capture the entire environment. Using the same App, the Raspberry Pi on the ceiling can be commanded to take an image. This captured image is displayed on the App along with the grid lines. These grid lines divide the image into 100 smaller images (with 10 rows and 10 columns).

The robot is a point in the 2D map obtained. Now the START (in Red color, on the mobile robot), END (in Green color), and the Traveling path (in blue color) along which the user wants the robot to travel are drawn on the canvas of the 2D map. This occupancy grid is sent to the raspberry pi on the robot using the POST file header of the webserver.

Next, as the image is received by the processor, an empty matrix is created to represent the 100 coordinates of the occupancy grid. Traversal across each piece of the square using a sliding window is done. With this window as an iterator, Dominant Color in each coordinate is checked. If the dominant color is red, then that co-ordinate is marked as START point in a list and if the dominant color is green, then that co-ordinate is marked as END in another list. Simultaneously, if the dominant color for any coordinate turns out blue, then those coordinates are stored in a TRAVEL list.

After acquiring these lists, the path planning function creates a list with all the coordinates in an ordered manner. After obtaining the planned path, this list is passed to the Localization function to perform the necessary movement of the rover.

After reaching the Destination, if the user wants a mobile robot to return to the initial starting point in the same manner, then the same planned path can be used. In such a situation, the elements in the planned path list are reversed and passed to the Localization function. This eliminates the need for re-planning the path.

VII. Limitations

Many limitations that have not been addressed in the present work and are considered as future works.

- Placing the markers and the line of the path on the generated map must be precise.
- The surface on which the rover will travel must be free of the excess mixture of color. The Presence of many colors will reduce the accuracy of the planned path.
- Lighting conditions while capturing the environment affects the accuracy of the planned path.
- The map doesn't account for the heights and depths of the surface.
- Initial Position of the mobile robot before starting the process must be placed facing UP with respect to the Image.
- The drawn line must never cross paths.

VIII. Further Improvisations

Existing algorithms such as Astar etc. can be used, after obtaining the start and endpoint, so that the convenience for the user can be increased. In this way, the user would not have to draw the path to be followed. But this comes with the expense of losing the freedom to obtain a user-desired path.

IX. CONCLUSION

In Conclusion, an experiment has been demonstrated employing the proposed path planning technique which increases the comfort of map generation and driving the robot in the desired path.

We conducted 27 test-runs with a target destination at distances ranging from 1m to 8m and achieved a success rate of 87% with a deviation of 2-3 cm. We plan to address the mentioned limitations in future work.

X. References

[1]. Gunasekaran, K. U., Krell, E., Sheta, A., & King, S. A. (2018), "Map Generation and Path Planning for Autonomous Mobile Robot in Static Environments Using GA., 8th International Conference on Computer Science and Information Technology (CSIT).

[2]. Mishrs, P., Sujith, B. S., & Mall, K. (2010), "A novel path planning algorithm for autonomous robot navigation", International Conference on Computer Applications and Industrial Electronics (ICCAIE 2010).

[3]. Avutu, S. R., Bhatia, D., & Reddy, B. V. (2017), "Design of touch screen based robot with obstacle detection module for autonomous path navigation", TENCON 2017 - 2017 IEEE Region 10 Conference.

[4]. Jitin Kumar Goyal, K.S. Nagla (2014), "New Approach of Path Planning for Mobile Robots", International Conference on Advances in Computing, Communications and Informatics (ICACCI)

[5]. Cherni, F., Boutereaa, Y., Rekik, C., & Derbel, N. (2015), "Autonomous mobile robot navigation algorithm for planning collision-free path designed in dynamic environments", 9th Jordanian International Electrical and Electronics Engineering Conference (JIEEEC).

[6]. Wang, J., & Chen, Z. (2018), "A Novel Hybrid Map Based Global Path Planning Method", 3rd Asia-Pacific Conference on Intelligent Robot Systems (ACIRS).

[7]. Ravankar, A., Ravankar, A. A., Kobayashi, Y., & Emaru, T. (2017), "Can robots help each other to plan optimal paths in dynamic maps", 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE).

[8]. Injarapu, A. S. H. H. V., & Gawre, S. K. (2017), "A survey of autonomous mobile robot path planning approaches", International Conference on Recent Innovations in Signal Processing and Embedded Systems (RISE).

[9]. Feng, Q., Gao, J., & Deng, X. (2016), "Path planner for UAVs navigation based on A* algorithm incorporating intersection", IEEE Chinese Guidance, Navigation and Control Conference (CGNCC).

[10]. Idrees, A., Iqbal, Z., & Ishfaq, M. (2015), "An efficient indoor navigation technique to find optimal route for blinds using QR codes", IEEE 10th Conference on Industrial Electronics and Applications

[11]. Jia-Heng Zhou, & Huei-Yung Lin. (2011), "A self-localization and path planning technique for mobile robot navigation", 9th World Congress on Intelligent Control and Automation.

[12]. Hassani, I., Maalej, I., & Rekik, C. (2018), "Robot Path Planning with Avoiding Obstacles in Known Environment Using Free Segments and Turning Points Algorithm", Mathematical Problems in Engineering, 1–13.