2019 4th International Electrical Engineering Conference (IEEC 2019)
Jan, 2019 at IEP Centre, Karachi, Pakistan

VISION BASED MULTIPLE ROBOTS AUTONOMOUS NAVIGATION SYSTEM USING LOCAL SENSORS

Sufyan Raza, Arsalan Qaisar*, Hunain Nasir*, Faizan Ellahi*, Asif memon* & Agha Shakir*

Institute of Industrial Electronics Engineering(IIEE)
Karachi, 75300, Pakistan (sufyanraza.26@iiee.edu.pk)* Corresponding author

Abstract: The objective of this research is to develop an overall fast, robust, consummate, & multiple robot autonomous navigation system, that is capable of selecting the nearest robot based on the destination, for it to traverse from its current node to the final, without any human interaction. This paper presents the approach utilizing computer vision method, which includes a top mounted camera capturing the overall environment, that the processor uses for the tag based localization and static mapping, afterwards planning an obstacle free path using adapted A-star algorithm of 4 adjacent nodes expansion, and eventually utilizing encoder motors, gyro sensor, & discrete camera feedback to ensure precise tracking. The obtained results were within minimal errors and could be further processed using either image stitching, for wide industrial area coverage, or it could be exploited by some dynamic robotic application**.**

Keywords: Image processing, computer vision, localization, autonomous navigation, path planning, path tracking, discrete camera feedback.

I. INTRODUCTION

The process of autonomous navigation was created for A.I based robotic applications [1], to enable the robot for its obstacle free maneuver in a given environment. A method was needed that could enable the robot to travel in an arena on its own. However, for the success of this method, a huge amount of information was needed by the robot, which became the cause for new adaptive solutions, utilizing fast algorithms, precise sensors, multiple feedbacks etc.

As time proceeded, the evolution of algorithms, through constant research, helped increase the overall data handling and processing speed. Information from multiple sensors was synchronized and processed within milliseconds[2], by the use of high speed controllers and processors, in turn providing better results, but still, the system lacked the overall speed, that it should had attained, which forced the researchers to pursue the study further.

By now, this moveable robot, either differential or omni-directional had sensors embedded within it, along with or with not the primary sensing element. The use of camera, lidar or radar type ranging instruments came into use with its data either being processed within the robot or through some external processor. The use of external processing in turn enabled wireless communication protocols and modules to be used within the system.

In addition to all of this hardware, more than one robot[3] had to be compensated in some way so as to make the system more application oriented. This stimulated the slave to slave communication or advance sensing concept for multiple robots to get synchronized within.

II. LITERATURE REVIEW

The cumbersome process of autonomous navigation was divided into multiple steps, including localization, mapping, path-planning and traversing, and each step had a number of solutions.

One study recorded by Zhaoxiang Li, and Gang Liu [4] involved utilizing GPS (global positioning system) to complete the process till localization, and for the purpose of feedback controlled tracking. The extraction of latitude and longitude was carried out using visual studio, and was compared with destination for further control instructions. The result had deviation both in terms of movement as well as reached node.

S. Shojaeipour et. Al.[5] used a webcam for autonomous Navigation. He placed the webcam on his robot and used the MATLAB image processing toolbox for planning an obstacle free path. He planned the path using the Cell decomposition method in which the position of obstacles was identified and cells corresponding to that location were eliminated and the path was planned from the remaining cells.

Eric Royer et. Al. [6] used a similar approach as he mounted a webcam on his vehicle, but it required the mapping of surrounding prior to the autonomous navigation. The mapping was done by manually navigating the vehicle around. The map was saved by the robot and was used later to plan the path from one point to another. The camera was used to dynamically localize the robot which was used for tracking of the planned path.

One of the most important step for Autonomous Navigation is path planning Norlida Buniyamin[7] presented a comparison of several Path planning algorithms and their pros and cons which were supported by various simulation results.

Another approach utilizing GPS and laser scanners was given by Y. Morales et. Al.[8] for outdoor pathway vehicles. The method employed built in map and results from various local sensors to process within laptop and generate the corresponding actuated signals.

H. Grewal[9] designed a wheelchair for patients suffering from arthritis or neuron degenerative diseases. It used a lidar sensor to map out the environment around the wheel chair. Lidar and encoder motor was also used to autonomously navigate from one point to another. ROS environment was used to extract and process the lidar and odometry data.

David J. F. Vaz et. Al. [10] designed a navigation system to navigate the robot over a predefined path. It used a GPS sensor, odometry data and data from IMU to successfully navigate the robot to its destination

III. METHODOLOGY

a. localization and mapping:

The camera was placed at the roof. Firstly an image was taken without the robots and obstacles. This image served as the reference to locate obstacles. For localization of the robots a tag was placed on both the vehicles. These tags are used to extract the position and orientation of the vehicle from the image. Since the tags were black and white, therefore it allowed us to eliminate the complexities of color image processing. For mapping out obstacles the image was compared to a reference image and the resultant image was divided into grids and averaged over a single grid. After averaging a grayscale threshold was set and image was digitized to 2 colors only. Since the vehicle consisted of several grids in the image therefore the grids adjacent to obstacles were also considered as obstacles. Finally the coordinates of vehicles were masked into the resultant image to produce a final map. This map included the position of both the robots and how obstacles were placed around the robots

b. path planning

Path planning for both the vehicles was done by using A* algorithm. For each node 4 of its adjacent nodes were considered as its neighbors so robots can only take 90 degrees turns. For the optimization of algorithm priority queue was used as the data structure for maintaining the adjacency list of the expanded nodes. As the A* algorithm tends to go for the node which minimizes its cost function, therefore it is really important to select your heuristic function to optimize the Algorithm. Manhattan distance was used as the heuristic function because the robot was allowed to move in four directions only. The shorter of the two path is conveyed to its corresponding robot. The path is split in such a way so the vehicle can move grid by grid.

c. path tracking

The real task was to improve the accuracy of path tracking with a considerable increment in the speed of vehicle for that purpose the Encoder DC motors were used for linear motion and to make sure that the vehicle is moving straight MPU6050 was used. When the vehicle started moving the initial value of the Gyro sensor was saved as a reference and using that value the deviation in the angle of vehicle was monitored. The speed of both the motors were varied corresponding to the deviations in the angle so as to keep the robot.

Moving in a straight line. For each 90 degrees turn the Gyro sensor tried to turn 90 degrees but due to the inaccuracy of Gyro it was rarely able to complete its 90 degrees turn. Therefore a discrete feedback from camera was taken after every attempt by the robot to turn a full 90 degrees and a control signal was generated proportional to the error which compensates for the inaccuracy of gyro sensor. During the path tracking if any dynamic obstacle was introduced into the system and it intercepted the path of robot then the robot stops and the process was repeated to plan a new path. The dynamic obstacle was identified by the ultrasonic sensor placed in front of robot.

IV. RESULTS

The images were obtained using IP camera and were processed on a Linux based system. Fig 1 depicts the original image which is to be processed. A reference image was taken prior to that image which excludes all the obstacles and robots.

Fig 1 Real time image prior to processing.

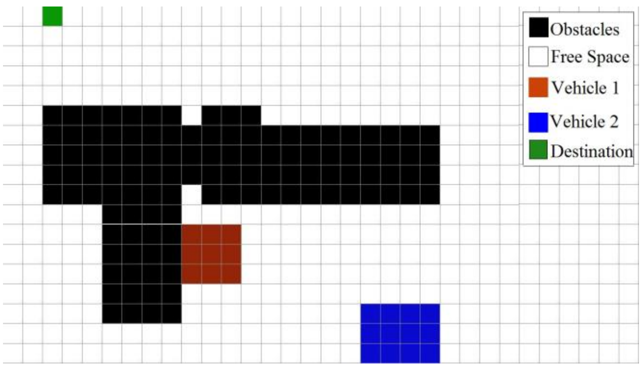

Fig. 2 Resultant image after localization and mapping.

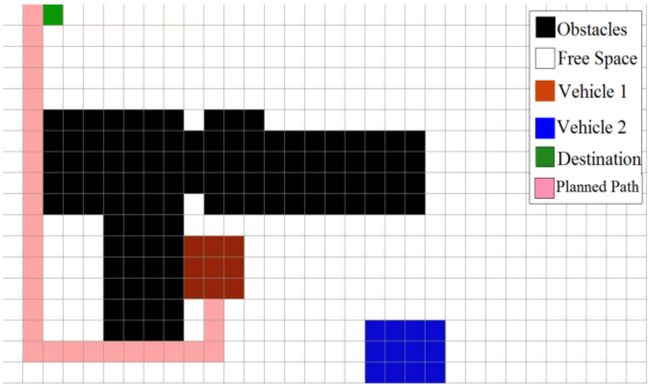

Fig. 3 Planned path for vehicle 1.

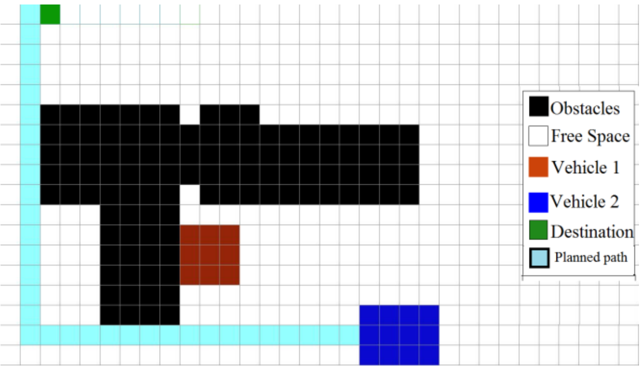

Fig. 4 path planning for vehicle 2.

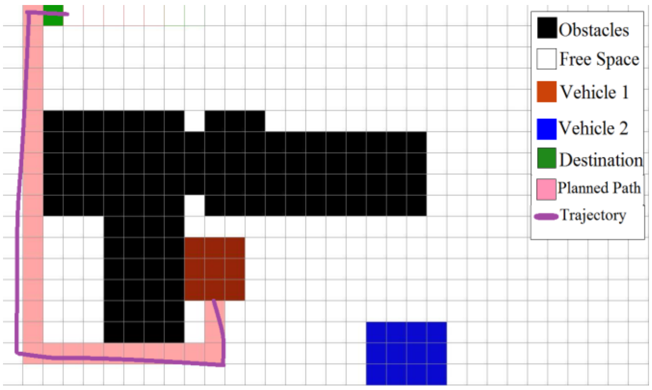

Fig. 5 path tracking for vehicle.

Fig 2 delineates the processed image after the localization and mapping process therefore it includes the position of obstacles and the robots. Fig 2 is further processed separately for both the robots. Fig 3 and Fig 4 delineates the planned path for robot 1 and robot 2 respectively. Since the path from Figure 3 is smaller therefore Robot 1 will transverse towards the destination. Fig 5 delineates the trajectory of the vehicle over the planned path. The trajectory of vehicle was mapped out by using video stream of roof mounted camera. Fig. 6 is the trajectory of path from the study of David J. F. Vaz et. Al.[10]. In comparison to our study it has the trajectory of robot over a larger distance. The length of rectangle in Fig 6 is 25m and width is 15m.The green line represents the ideal path whereas the red line represents the actual trajectory.

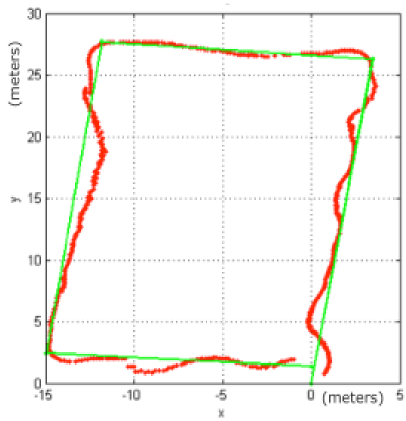

Fig.6 Trajectory of path using Kalman Filter algorithm [10]

Table 1 Deviation of trajectory from planned path.

Cell Number	Deviation from Desired position (cm)
1	0.75
2	3
3	3.375
4	4
5	3.5
6	3.125
7	2.625
8	2.125
9	2
10	1.75
12	1.625
13	1.625
14	0.625
15	4.8
16	5.32
17	4.5
18	4.75
19	4.75
20	4.275
21	4.15
22	4
23	3.378
24	3.378
25	4.25
26	3.875
27	3.375
28	2.875
29	2.375
30	2
31	1.5
32	0.625

Table 1 represents the deviation in the planned path for each grid that was present in the planned path. The deviation or error is measured form the center of the grid.

Fig 7 and Fig. 8 shows the deviation chart or the absolute difference of the robot from the ideal position for this study and the study made by David J. F. Vaz et. Al.[10] respectively. Although the latter corresponds to the experiment done on a large scale, there is a considerable difference in the accuracy of both the approaches. The results produced from the approach presented in this paper has the maximum deviation of 5.32 cm from desired position. Whereas Fig 8 shows that the maximum error is 130cm in the approach adopted by David J. F. Vaz et. Al

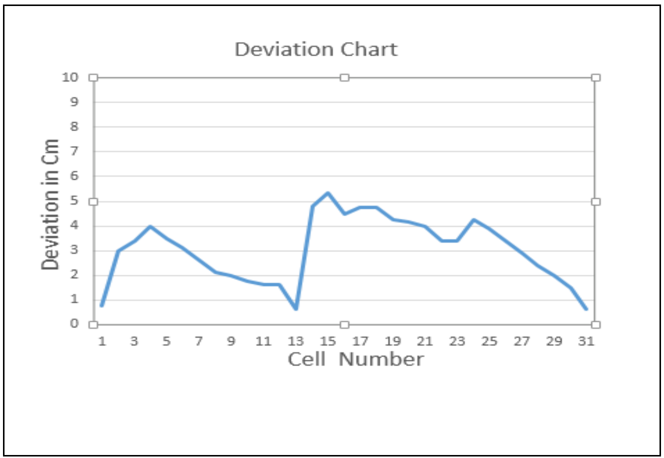

Fig. 7 Graphical representation of Deviation in trajectory.

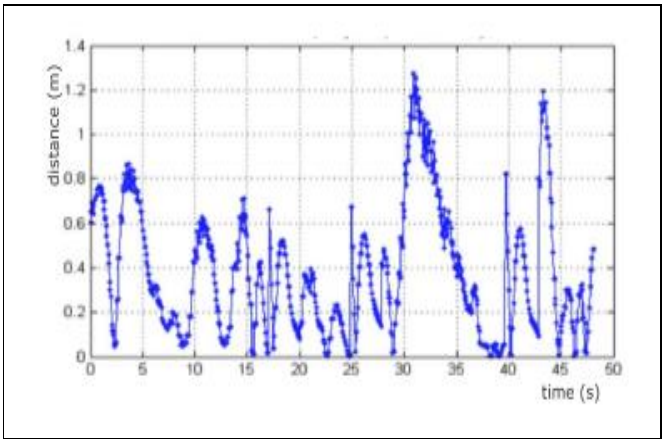

Fig. 8 Deviation chart for David J. F. Vaz et. al.[10]study

V. CONCLUSION

The main aim of this study was to improve the speed of system without considerably effecting the accuracy of the system which was achieved by the feedback from local sensors instead of continuous feedback from overhead camera. The IMU sensor and encoder motors were used as the local sensors. The results obtained from this configuration were reliable. A comparison was made with an existing approach which further reaffirms the reliability of the approach. Further improvements can be made by replacing the DC encoder motors with stepper motors. The vision of camera limits the area of operation of robots to a certain region which can be increased by using multiple cameras and then using image stitching algorithm to eventually end up with a single image to be processed.

REFERENCES

[1] https://en.wikipedia.org/wiki/Autonomous_robot

[2] Sanjiv Singh, Paul Keller, “Obstacle Detection for High Speed Autonomous Navigation,” *International Conference on Robotics and Automation*, April 1991.

[3] F. Mondada, A. Guignard, M. Bonani, D. Bar, M. Lauria and D. Floreano, "SWARM-BOT: from concept to implementation," *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, Las Vegas, NV, USA, 2003, pp. 1626-1631 vol.2.

[4] Zhaoxiang Li, Gang Liu, “Autonomous Navigation System Based on GPS,” *Computer And Computing Technologies In Agriculture, vol. II*, pp. 943-950, 2007.

[5] S. Shojaeipour, S. M. Haris, K. Khalili and A. Shojaeipour, "Motion planning for mobile robot navigation using combine Quad-Tree Decomposition and Voronoi Diagrams," *The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, pp. 90-93, 2010.

[6] Eric Royer, Maxime Lhuillier, Micheal Dhome, Jean-Marc Lavis, “Monocular Vision for Mobile Robot Localization and Autonomous Navigation,” *International Journal of Computer Vision, vol. 74*, pp. 237-260, 2007.

[7] E. J. Gómez, F. M. Santa and F. H. M. Sarmiento, "A comparative study of geometric path planning methods for a mobile robot: Potential field and voronoi diagrams," *2013 II International Congress of Engineering Mechatronics and Automation (CIIMA)*, Bogota, pp. 1-6, 2013.

[8] Morales, Y & Carballo, Alexander & Takeuchi, Eijiro & Aburadani, Atsushi & Tsubouchi, Takashi. “Autonomous robot”96/ navigation in outdoor pedestrian walkways,” *J. Field Robotics*, 2009.

[9] H. Grewal, A. Matthews, R. Tea and K. George, "LIDAR-based autonomous wheelchair," *IEEE Sensors Applications Symposium (SAS)*, Glassboro, NJ, pp. 1-6. 2017.

[10] D. J. F. Vaz, F. A. V. João, A. J. Serralheiro, A. J. Serralheiro, J. A. B. Gerald and J. A. B. Gerald, "Autopilot — An autonomous navigation system," *2016 Conference on Design of Circuits and Integrated Systems (DCIS)*, Granada, 2016, pp. 1-5.