



Research Article

Self-Localization of Mobile Robot Using Map Matching Algorithm

Bushra Shaikh, Sadia Muniza Faraz*, Syed Riaz un Nabi Jafri, Syed Usman Ali

Department of Electronic Engineering, NED University of Engineering and Technology, 75270, Karachi, Pakistan.

Abstract: The use of mobile robots is increasing day by day. However, an effective robotic navigation system demands efficient localization and path planning techniques. A map matching algorithm based on matrix matching is being proposed in this paper. The localization of robot is determined by the path acquired during the random maneuvering on the grid based maze, whereas, the shortest path to the target position is determined by estimating x and y coordinates. The algorithm interprets the map of the environment, expressed by grid based model and generates an optimal or near optimal path. The proposed algorithm is demonstrated by simulation studies using MATLAB and VREP. The algorithm is verified by a sensors based robotic setup with three different maps. The experimental results verified self-localization technique with an accuracy of 98 % for a mobile robot prototype in an indoor environment. The proposed algorithm is simple, low cost, independent of lighting conditions and requires less computational steps to identify the exact location on the map.

Received: August 17, 2020; **Accepted:** June 12, 2021; **Published:** June 30, 2021

***Correspondence:** Sadia Muniza Faraz, Department of Electronic Engineering, NED University of Engineering and Technology, 75270, Karachi, Pakistan; **Email:** smuniza@neduet.edu.pk

Citation: Shaikh, B., S.M. Faraz, S.R.N. Jafri, S.U. Ali. 2021. Self-localization of mobile robot using map matching algorithm. *Journal of Engineering and Applied Sciences*, 40(1): 69-77.

DOI: <https://dx.doi.org/10.17582/journal.jeas/40.1.69.77>

Keywords: Localization, Map matching, Mobile robot

Introduction

Mobile robots have achieved significant success since last three decades. They assist with several work processes and tasks in outdoor and indoor environments such as mine exploration, videography, industrial manufacturing, medical procedures, space exploration, agriculture and unmanned or autonomous vehicles (Fragapane *et al.*, 2020; Song *et al.*, 2018; De Ryck *et al.*, 2020; Kato *et al.*, 2018).

For travelling autonomously in an unknown environment, efficient and accurate navigation mechanism is essential for many safety applications. Therefore, localization and optimum path planning of mobile robots involving extraction of meaningful data from sensors, estimation of correct location, planning of necessary steps to the target and achieving the

desired trajectory is important (Panigrahi and Bisoy, 2021).

The techniques and systems for self-localization estimation are based on wheel odometry, global navigation satellite system (GNSS), global position system (GPS), inertial navigation system (INS), and visual odometry (VO) etc. However, each technique has its own strengths and weaknesses. Such as wheel odometry has been the most popular self-localization method because of its simplicity and cost effectiveness, but perform poorly in uneven, slippery and complex terrain (Fernandez and Price, 2004). Global Navigation Satellite System (GNSS) is widely used but GNSS data can only be acquired outdoor and its localization precision is inadequate. Likewise, the Global Positioning System (GPS) can provide absolute position but the signals are mostly

deteriorated in the urban environment (Okawa, 2019; Wegener and Schnieder, 2012). High precision visual inertial navigation system (INS) is expensive and prone to drift accumulation (Leutenegger *et al.*, 2015).

Self-localization is one of the main part of autonomous robots for collision avoidance while moving in a facility (Sobreira *et al.*, 2019). In fact, localization and mapping are dependent on one another and map-matching is by far the most popular approach for self-localization estimation. Self-localization and mapping both can be collectively achieved by Simultaneous Localization and Mapping (SLAM), proposed by R.C. Smith and further modified by M.G. Dissanayake (Smith and Cheeseman, 1986; Smith *et al.*, 1990; Dissanayake *et al.*, 2001). SLAM algorithm is based on the representation and estimation of spatial uncertainty. In this method the already stored map of any vicinity is matched with the data retrieved from the sensors. By using the data of joints and relative positioning of the parts, Chen *et al.* (2018) designed disaster response robots (DRRs). These DRRs performed SLAM task for terrain exploration in low light conditions with less number of sensors. The success of SLAM based self-driving cars have drawn the worldwide attention and SLAM based robots have attracted the mass market for home automation (Montemerlo *et al.*, 2008; Knight, 2015).

In filtering techniques, Kalman filter is a commonly used technique for estimating vehicle positions. Jawanmardi *et al.* (2020) proposed a method in which the input scan of the LiDAR is matched to the prebuilt map to get high accuracy position of the vehicle. They identified error sources in self-localization based on map matching and proposed a framework for pre-estimation of map-matching error. Extended Kalman Filter (EKF) and Particle Filter (Smith and Cheeseman, 1986) are two popular estimation methods commonly used in the SLAM algorithm (Aria, 2019). Yu *et al.* (2019) proposed an EKF and MM based hybrid dead reckoning error correction scheme to improve positioning accuracy for vehicle self-localization. Placin *et al.* (2020) employed an on board tilted scanning strategy of the sensors for the self-localization of mobile robots. The tilted down (push broom application) application helped in detecting and avoiding low-height obstacles and slopes in trajectory of mobile robot.

In another study Wang and Takahashi (2018) developed a low cost infrared LED emitters and receivers based system for the self-localization of mobile robots in indoor environment. The proposed system also solved the problem of limited IDs by using a combination of unique ID encoding landmarks and repeated ID encoding landmarks. Fu *et al.* (2007) designed an autonomous mobile robot SLAM system using laser range finders as primary sensors to collect information for indoor environment. The visual information received from laser range finder and CCD camera were found to be credible for recognition and judgement. For sparse natured problems, graph-based optimization techniques may be employed to solve the problem. Where a factor graph is used which represents variables and measurements between the nodes. Node and edge sparsification technique is also used to reduce overall size of graph or map by eliminating less informative nodes hence the overall computation complexity is reduced (Mazuran *et al.*, 2014).

In this work, a solution for map matching and self-localization in the indoor environment is proposed based on matrix matching. The problem may be framed in the context of obtaining simple, low cost, less computationally demanding solution to interprets the map of the environment, expressed by grid based model and generate an optimal or near optimal path. The proposed algorithm makes use of Nonlinear Graph Sparsification (NGS) method. The generated path is compared with the complete map of environment and the position of bot is determined with reference to the targeted position. The technique is demonstrated by simulation studies using MATLAB and VREP and verified by prototype robotic setup with three different maps. The proposed solution is a low cost sensor based definite solution with no conditional dependency on the lightening conditions.

This paper is organized as follows: A brief introduction is given in this section. In section-2 we have described model design, hardware design and kinematic model used in simulation work. Section 3 outlines the simulation setup used to verify the proposed model, trajectory generation and steps involved in map matching. Section 4 presents real experiments to demonstrate the working of the proposed algorithm. The results are compared in this section. Finally, Section 5 concludes this paper.

Experimental setup design and modelling

Robot model design: To simulate and verify the proposed algorithm a line following robot model was designed in VREP to mimic the real world. The robot model was designed with three pairs of Infrared (IR) sensors to differentiate white surface and black line. Three additional sensors were attached on front, left and right sides for detecting T - junctions and + junctions. For differential drive mode, each wheel was connected to independent motor and turning effect was achieved with speed variations of wheels. The block diagram and final model with sensors, motors and on board controller is shown in Figure 1a, b. PID controller was used to achieve smooth motion. Position of robot on the line was estimated with the help of three IR sensors attached in the middle. At first three constant K_p , K_i and K_d were saved in the EEPROM of the controller. After initialization the robot was made to move on the line. The position feedback of robot with respect to line was obtained from IR sensors. Based on the feedback signal, auto set position and gain values saved in EEPROM, the PID loop generates PWM signal to the motors to drive the robot on the black line.

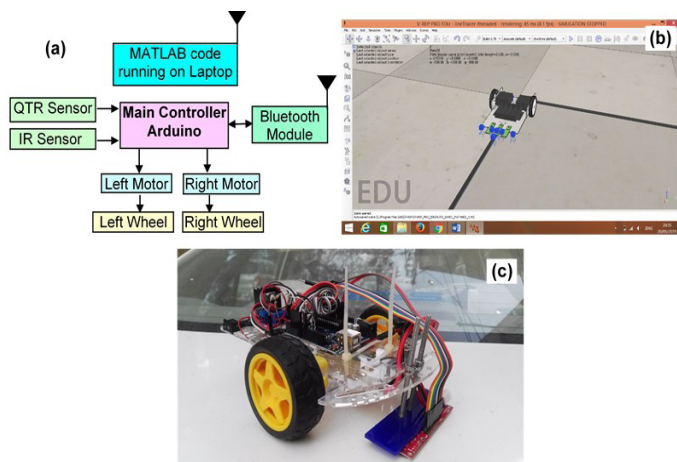


Figure 1: (a) Block diagram (b) Model developed in VREP (c) Prototype of Robot for algorithm verification.

Robot hardware design

To experimentally validate the algorithm a prototype of mobile robot has been developed. The image of robot is shown in Figure 1c. The model contains analogue QTR sensor array for the line following propose. Additional three digital IR sensor are used at the left, front and right side of the robot in order to detect the T-junction or + junction. The movement of robot is based on the differential drive mechanism. Bluetooth module HC-05 is used to establish wireless connection between the Arduino and MATLAB.

The mechanism used in VREP simulations is deployed in to the prototype model. After sensor calibration and initialization steps, the robot first generate trajectory in MATLAB workspace, after completion of map matching algorithm, MATLAB execute the path planning program. Once, the shortest path is calculated, robot starts moving on arena, whenever robot find the interception of multiple path, it generates an interrupt signal. On receiving the interrupt signal MATLAB transmits back the turn signal associated with the shortest path.

Kinematic model

The position of mobile robot is represented by its coordinates and Orientation. Its position in the world frame is represented by the coordinates of centre of mass i.e., X, Y, θ . Where X and Y represent the distance of robot from the reference frame and θ represents the robot pose as shown in Figure 2. The control input and speed of wheels are related as $V_R = r \times \omega_R$ and $V_L = r \times \omega_L$.

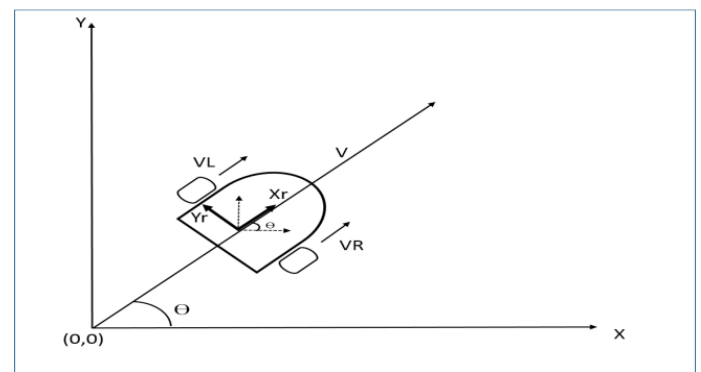


Figure 2: Kinematic model of differential drive model.

Where r is the radius of wheel, V_L and V_R are left and right wheel velocities and ω_L and ω_R are angular velocities of left and right wheels respectively. The kinematic model is based on the assumption that robot is moving on the plan surface, i.e., it does not include pitch or roll component and only rotate along Z-axis. The linear velocity(V) and angular velocity(ω) of robot are given by Equations 1 and 2.

$$n = \frac{V_R + V_L}{2} \dots(1)$$

$$w = \frac{V_R - V_L}{d} \dots(2)$$

Where, d is the distance between wheels. The tangential velocities along x axis and y axis are given by Equations 3 and 4.

$$\dot{X} = v \cos q \quad \dots(3)$$

$$\dot{Y} = v \sin q \quad \dots(4)$$

If the robot moves with the linear velocity of v and angular velocity of ω than the state matrix can be represented as $[\dot{X} \dot{Y} \dot{\theta}]$ given by;

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos q & 0 \\ \sin q & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad \dots(5)$$

The finalized kinematic model of differential drive robot is obtained from Equations 3, 4 and 5, as given below.

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 2v \cos q \\ 2v \sin q \\ d \omega \end{bmatrix} \begin{bmatrix} w_R \\ w_L \end{bmatrix} \quad \dots(6)$$

This simulation work in this research is based on the kinematic model obtained in Equation 6.

Materials and Methods

To drive the robot over the black line, basic code was written in threaded child scripts. The main program was written in MATLAB. The VREP model and scene were directly controlled by external applications with the help of application programming Interface (API). After establishing the connection, VREP and MATLAB communicated in the form of signals. The basic interfacing of MATLAB and VREP is shown in Figure 3.

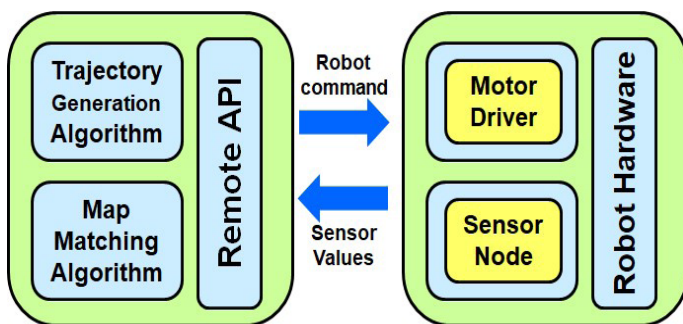


Figure 3: Interfacing MATLAB and VREP.

Trajectory generation

In the beginning, robot has no prior knowledge of its location. When it starts moving in an initial direction then VREP generates specific signals corresponding to its movements. The path is updated if the robot takes left or right turn however it remains unchanged if moves straight. The covered trajectory is imported in MATLAB workspace for trajectory generation. A free running counter is monitored and the robot is stopped generating an interrupt signal, when a particular time is met. The trajectory generation process is shown by the flow chart in Figure 4.

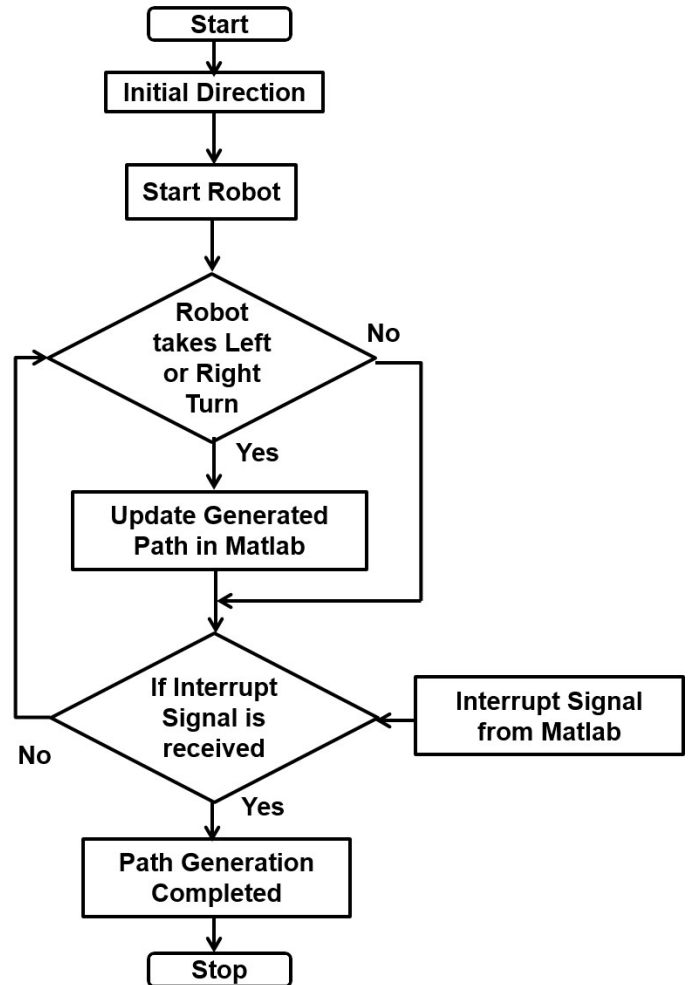


Figure 4: Trajectory Generation in MATLAB.

Map matching for self localization

Self-localization is being done with the help of map matching algorithm for use in an indoor environment with no obstacles, within the defined range of allowed path. It makes use of lane markings or black lines on the floor, as the guided or allowed path to travel. The position estimation is done by matching feature of the generated sample map with the actual map of complete field. This approach allows the position estimation of robot with higher degree of precision. A

simple pattern and its trajectory in VREP are shown in Figure 5a, b.

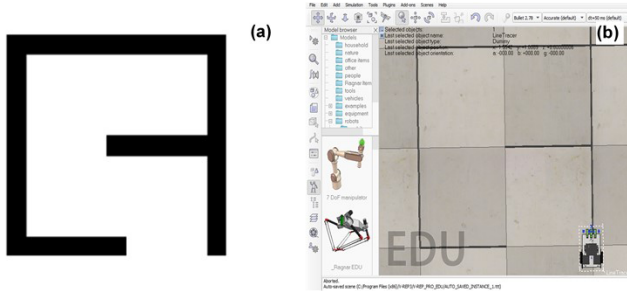


Figure 5: (a) Generated pattern in MATLAB (b) Trajectory covered in VREP.

The specific pattern of data is recognized by comparing generated and segmented patterns. In this comparison the generated and segmented portion of whole image is thought of as sliding window rolling over the complete picture with predefined steps. It is performed row wise and after the completion of one row the window rolls back to the first column of next row. A pattern of order of 7×7 is compared with the same sized segment of main image and as a result an output matrix of order 5×5 is generated as shown in Figure 6a. The entities of this matrix, can be considered as the degree of similarity amongst the two, with 0 being non similarity and 1 means, both matrices are 100 percent similar. Pattern recognition techniques were used to find the correlation between the generated trajectory and the global map. Once, the data is gathered in the form of distance vector, it is first converted into a grey image. This image is then compared with the equal sized portion of global map. The algorithms mark the current segment as matched, if the similarity check is approximately 98.99 percent or greater.

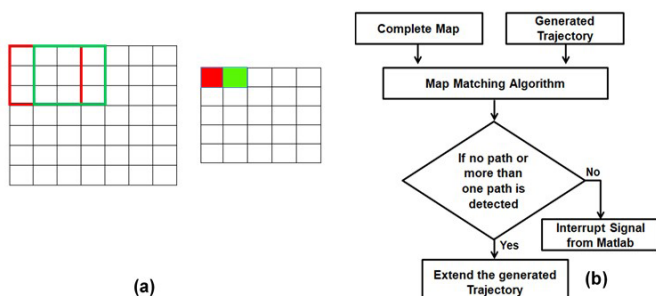


Figure 6: (a) Visualization of 3×3 convolutions around an input (b) Complete map matching algorithm.

The overall process is shown in flow chart in Figure 6b. The algorithm starts with the input of the already recorded map and the generated trajectory fed the map

mapping algorithm. Algorithm processed inputs on the basis of the algorithm running through MATLAB code. If the coordinates of both maps are matched, then program directs the main controller to guide the robot motors. The map matching algorithm involved in the decision process Figure 6b is shown in Figure 7. In this process, at first an imaginary window of size $m \times n$ is considered over the upper left corner of the complete map. Where, m and n are the dimensions of the generated grey image. Then map-matching algorithm computes the similarity between the patterns formed by the covered trajectory against the same sized portion of the global map. If the similarity is greater than 98 % then selected portion of the global map is marked as the estimated position. If the similarity is less than 98 % then imaginary window in moved one step forward and again the similarity is computed. This cycle continues until one selected portion is matched with the generated trajectory. On the basis of this map-matching algorithm the robot confirms its position and will localized itself in that environment.

Results and Discussion

The proposed algorithm and program has been tested and verified using two wheeled differential drive robot for a map of 35, 40 and 90 cells map as shown in Figure 8a, b, c. Path is generated in forms of small segments and robot moves over black line. The robot stops after completing approximately two segments and lets main controller execute the map matching algorithm.

The occurrence of same pattern at multiple places may lead to false judgement of position. This is avoided by checking the number of matched patterns and if the central controller returns more than one pair of coordinates, the robot continues to move over the path for the allocated time. The complete program is re-executed by the controller for searching the extended pattern. This cycle continues, unless a unique pair of coordinates is obtained. The shaded area represents the estimated position of robot as shown in Figure 8. The number of iterations and execution times observed in the experimental run are listed in Table 1. For a sample trajectory, it is observed that in the very start the algorithm took large number of iterations and execution time in processing the maps and giving the exact location. However, the number of iterations and execution time is decreased in identifying the location when the number of cells are increased. Although, large trajectories take longer time.

Table 1: Computational time and number of iterations performed by the algorithm.

No. of cells, covered in the sample trajectory	35 cells map		40 cells map		90 cells map	
	Iterations	Execution time (sec)	Iterations	Execution time (sec)	Iterations	Execution time (sec)
1 cell	-	-	737	12.1	-	-
2 cells	582	10	622	10.1	617	10.3
4 cells	437	7.3	367	6.4	412	6.8
6 cells	385	7.04	362	7.3	207	3.5
8 cells	302	6.3	-	-	223	6.5

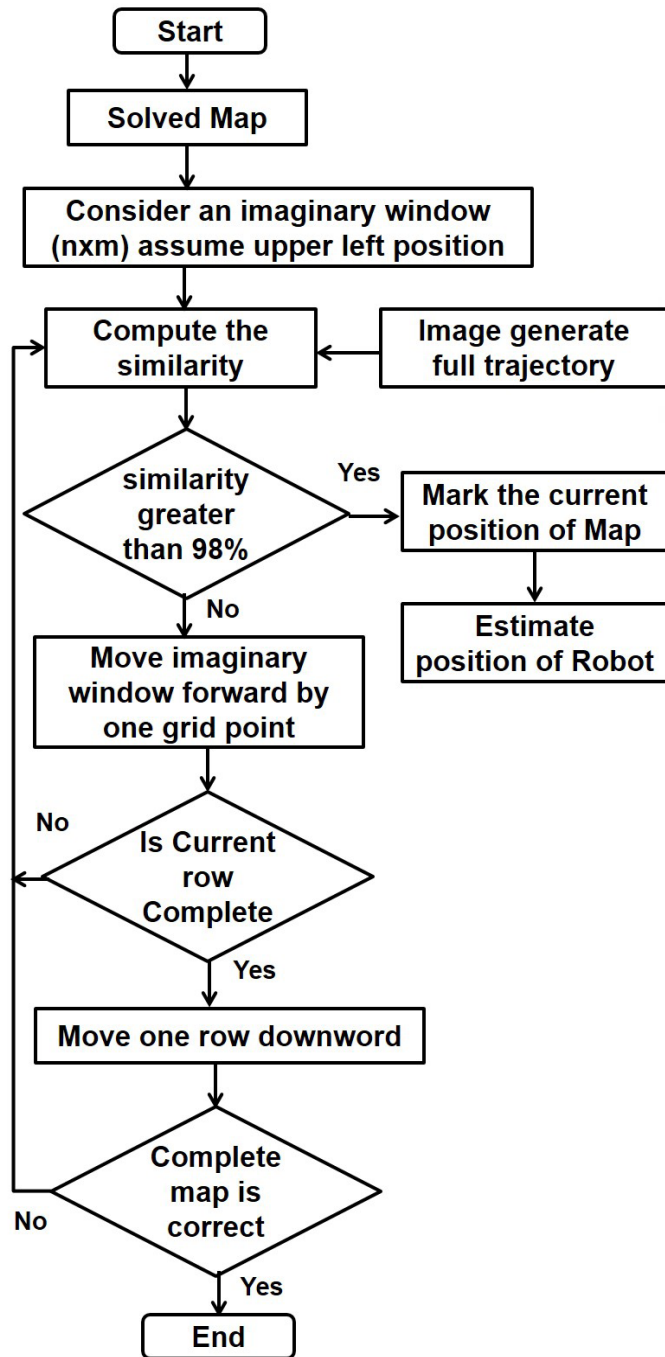


Figure 7: Map Matching Algorithm.

The map matching percentage is set to 98% therefore the marking of the position will be executed only when the location maps and the generated maps are 98% matched. Otherwise, the matrix window will

shift from the location and search for other location. Thus, an accuracy of 98% is achievable in our low cost and simple computational algorithm for localization in the indoor environment.

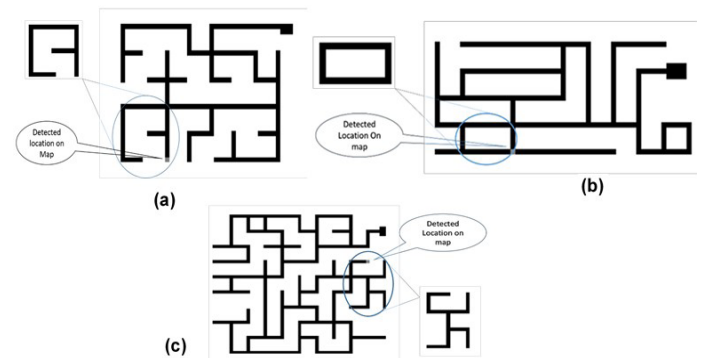


Figure 8: Localization of robot in MATLAB using map matching (a) 35 (b) 40 (c) 90 cells.

These results show the improvement in the accuracy and computations compared to artificial landmark recognition reported in literature (Zhong et al., 2017), where the system is based on camera and identification of the landmarks depends upon the light conditions. In the presence of artificial landmarks, our proposed system can perform accurately in modest light conditions. Secondly, self-localization of a robot in an environment depends on algorithm as well as the data quality. Accurate estimation of correct path in indoor environment using refinement technique based on probabilistic approach (Vivacqua et al., 2017) may result in collision in case diverting from the main path. It requires higher sensorial data load used at superior computational scheme. However, the proposed scheme of this paper in presence of artificial tracks works on lower sensorial feed with very light computational algorithm as compare to the probabilistic localization approach. In our proposed algorithm the position estimation is done by matching the features of generated sample map with the actual map of the vicinity. This algorithm has solved the problem of local similarity of map as raised in other research studies (Xia and Tang, 2019).

In our work, features are converted to image file which is used by algorithm to confirm location of robot in the environment by matching it with generated sample of map. Although, rapid localization systems or probabilistic approach based on Bayes Filter (Chia *et al.*, 2020; Maniscalco *et al.*, 2017) are accurate but computationally expensive, where conversion of the entire field into the geometrical coordinates demands more computational power for running the algorithm. By using defined tracks, our proposed algorithm involves less computation procedures with satisfactory results.

In this research study a computational solution has been proposed for map-matching in the indoor environment to localize the robot in presence of pre-defined tracks. The purpose of this work is to propose a low cost and a less computationally demanding solution for map-matching and localization as compared to the already proposed artificial intelligence based algorithms (Jin *et al.*, 2018). These algorithms are although well accurate but they are needed additional computational resources. Similarly, in another research study (Li *et al.*, 2017), the accuracy has been proved for map-matching using neural networks but computational load is high compared to the presented solution and algorithm which is using simple matrix matching technique for the patterns of the map.

Conclusions and Recommendations

An algorithm is proposed for self-localization and map matching of mobile robots in indoor environment. Unlike other camera based algorithms, our proposed solution works on the direct sensorial feed. The complete navigation algorithm is demonstrated by simulation studies and verified on a prototype using three different maps. The position of robot is identified with 98 % accuracy. The proposed solution is low cost as compare to other expensive visualization process based solutions having conditional dependency on the lightening conditions. Because of its simplicity, it may be implemented easily as compared to other computationally demanding algorithms requiring more steps and additional computational resources. The algorithm can further be improved by introducing advanced machine learning techniques to enhance its accuracy for use in the outdoor environment.

Novelty Statement

In this paper, a simple and low cost algorithm is proposed which is based on an existing and proven algorithm. The proposed algorithm is independent of lighting conditions and requires less computational steps to identify the exact location on the map.

Author's Contribution

The implementation and simulations were carried out by Bushra Shaikh. Technical review, verification, manuscript drafting and revisions were carried out by Sadia Muniza Faraz. Implementation results were validated by Syed Riaz un Nabi with reference to the current literature. Conceptualization and over all supervision for this research was done by Syed Usman Ali.

Conflict of interest

The authors have declared no conflict of interest.

References

- Aria, M., 2019. Real-time 2D mapping and localization algorithms for mobile robot applications. In IOP conference series: Materials science and engineering, 18- July 2019, Bandung, Indonesia, 662(2): 022131. <https://doi.org/10.1088/1757-899X/662/2/022131>
- Chen, K., M. Kamezaki, T. Katano, T. Kaneko, K. Azuma, T. Ishida, M. Seki, K. Ichiryu and S. Sugano. 2018. Compound locomotion control system combining crawling and walking for multi-crawler multi-arm robot to adapt unstructured and unknown terrain. *Robomech J.*, 5(1): 1-17. <https://doi.org/10.1186/s40648-018-0099-5>
- Chia, T.L., S.Y. Chiang and C.H. Hsieh. 2020. Rapid self-localization of robot based on omnidirectional vision technology. *Mach. Vis. Appl.*, 31(7): 1-20. <https://doi.org/10.1007/s00138-020-01129-7>
- De-Ryck, M., M. Versteyhe and F. Debrouwere. 2020. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *J. Manufact. Syst.*, 54: 152-173. <https://doi.org/10.1016/j.jmsy.2019.12.002>
- Dissanayake, M.G., P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba. 2001. A solution to the simultaneous localization and

- map building (SLAM) problem. *IEEE Trans. Robot. Automat.*, 17(3): 229-241. <https://doi.org/10.1109/70.938381>
- Fernandez, D., and A. Price. 2004. Visual odometry for an outdoor mobile robot. In: Anonymous 2004 IEEE Conference on robotics, automation and mechatronics., 1-3 Dec. 2004, Singapore, pp. 816-821.
- Fragapane, G., D. Ivanov, M. Peron, F. Sgarbossa and J.O. Strandhagen. 2020. Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics. *Ann. Operat. Res.*, pp. 1-19. <https://doi.org/10.1007/s10479-020-03526-7>
- Fu, S., H.Y. Liu, L.F. Gao and Y.X. Gai. 2007. SLAM for mobile robots using laser range finder and monocular vision. In 2007 14th International Conference on Mechatronics and Machine Vision in Practice, 4-6 Dec. 2007, Xiamen, China, pp. 91-96.
- Javanmardi, E., M. Javanmardi, Y. Gu and S. Kamijo. 2020. Pre-estimating self-localization error of NDT-based map-matching from map only. *IEEE Trans. Intell. Transport. Syst.*, pp. 1-15. <https://doi.org/10.1109/TITS.2020.3006854>
- Jin, X.B., T.L. Su, J.L. Kong, Y.T. Bai, B.B. Miao and C. Dou. 2018. State of the art mobile intelligence: Enabling robots to move like humans by estimating mobility with artificial intelligence. *Appl. Sci.*, 8(3): 379. <https://doi.org/10.3390/app8030379>
- Kato, S., S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii and T. Azumi. 2018. Autoware on board: Enabling autonomous vehicles with embedded systems. In 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems, Porto, Portugal, 11-13 April 2018, pp. 287-296. <https://doi.org/10.1109/ICCPS.2018.00035>
- Knight, W., 2015. The Roomba now sees and maps a home. *MIT Technology Review*. 2015.
- Leutenegger, S., S. Lynen, M. Bosse, R. Siegwart and P. Furgale. 2015. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.*, 34(3): 314-334. <https://doi.org/10.1177/0278364914554813>
- Li, J., H. Zhan, B.M. Chen, I. Reid and G.H. Lee. 2017. Deep learning for 2D scan matching and loop closure. In 2017 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), 24-28 Sept. 2017, Vancouver, BC, Canada, pp. 763-768. <https://doi.org/10.1109/IROS.2017.8202236>
- Maniscalco, U., I. Infantino and A. Manfre. 2017. Robust mobile robot self-localization by soft sensor paradigm. In 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), 5-7 Oct. 2017, Ottawa, ON, Canada, pp. 19-24. <https://doi.org/10.1109/IRIS.2017.8250092>
- Mazuran, M., G.D. Tipaldi, L. Spinello and W. Burgard. 2014. Nonlinear graph sparsification for SLAM. In *Robotics: Science and Systems. Proc. Robot. Sci. Syst.* 12 - 16 July, 2014, Berkeley, USA. pp. 1-8. <https://doi.org/10.15607/RSS.2014.X.040>
- Montemerlo, M., J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke and D. Johnston. 2008. Junior: The stanford entry in the urban challenge. *J. Field Robot.*, 25(9): 569-597. <https://doi.org/10.1002/rob.20258>
- Okawa, K., 2019. Self-localization estimation for mobile robot based on map-matching using downhill simplex method. *J. Robot. Mechat.*, 31(2): 212-220. <https://doi.org/10.20965/jrm.2019.p0212>
- Palacín, J., D. Martínez, E. Rubies and E. Clotet. 2020. Mobile robot self-localization with 2D push-broom LIDAR in a 2D map. *Sensors*, 20(9): 2500. <https://doi.org/10.3390/s20092500>
- Panigrahi, P.K. and S.K. Bisoy. 2021. Localization strategies for autonomous mobile robots: A review. *J. King Saud Univ. Comp. Inf. Sci.*, <https://doi.org/10.1016/j.jksuci.2021.02.015>
- Smith, R., M. Self and P. Cheeseman. 1990. Estimating uncertain spatial relationships in robotics. *Auto. Robot Vehicles*, Springer, New York. pp.167-193. https://doi.org/10.1007/978-1-4613-8997-2_14
- Smith, R.C., and P. Cheeseman. 1986. On the representation and estimation of spatial uncertainty. *Int. J. Robot. Res.*, 5(4): 56-68. <https://doi.org/10.1177/027836498600500404>
- Sobreira, H.M., A.P. Moreira, P.J. Costa, P.C.M.A. Farias, J. Lima, L.F. Rocha, I. Sousa and C.M. Costa. 2019. Map-matching algorithms for robot self-localization: A comparison between perfect match, iterative closest point and normal distributions transform. *Intell. Robot. Syst.*, 93: 533-546. <https://doi.org/10.1007/>

s10846-017-0765-5

- Song, J., J. Wang, L. Zhao, S. Huang and G. Dissanayake. 2018. MIS-SLAM: Real-time large-scale dense deformable SLAM system in minimal invasive surgery based on heterogeneous computing. *IEEE Robot. Automat. Lett.*, 3(4): 4068-4075. <https://doi.org/10.1109/LRA.2018.2856519>
- Vivacqua, R.P.D., M. Bertozzi, P. Cerri, F.N. Martins and R.F. Vassallo. 2017. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Trans. Intell. Transp. Syst.*, 19(2): 582-597. <https://doi.org/10.1109/TITS.2017.2752461>
- Wang, J. and Y. Takahashi. 2018. Indoor mobile robot self-localization based on a low-cost light system with a novel emitter arrangement. *ROBOMECH J.*, 5(1): 1-17. <https://doi.org/10.1186/s40648-018-0114-x>
- Wegener, M. and Schnieder, E., 2012. A measurement standard for vehicle localization and its ISO-compliant measurement uncertainty evaluation. *IEEE Trans. Instr. Measur.*, 61(11): 3003-3013. <https://doi.org/10.1109/TIM.2012.2203872>
- Xia, Z. and S. Tang. 2019. Robust self-localization system based on multi-sensor information fusion in city environments. In 2019 International Conference on Information Technology and Computer Application (ITCA), 20-22 Dec. 2019, Guangzhou, China, pp. 14-18. <https://doi.org/10.1109/ITCA49981.2019.00011>
- Yu, B., L. Dong, D. Xue, H. Zhu, X. Geng, R. Huang and J. Wang. 2019. A hybrid dead reckoning error correction scheme based on extended Kalman filter and map matching for vehicle self-localization. *J. Intell. Transp. Syst.*, 23(1): 84-98. <https://doi.org/10.1080/15472450.2018.1527693>
- Zhong, X., Y. Zhou and H. Liu. 2017. Design and recognition of artificial landmarks for reliable indoor self-localization of mobile robots. *Int. J. Adv. Robot. Syst.*, 14(1): 1-13. <https://doi.org/10.1177/1729881417693489>