e-ISSN 2320 –7876 www.ijfans.org
Vol.10, Iss.03, 2021
Research Paper UGC CARE Listed (Group -I) Journal

Improving 3D Mobile Robot Navigation: Utilizing DQMD-DNN For Enhanced Motion Planning And 3D Mapping

Revannath B. Kakade[1*]
[1*]Lecturer in Dept. of Electrical Engg.Government residential women's polytechnic Latur, 413532 (M.S.), India.

***Corresponding Author:**Revannath B. Kakade
***Lecturer in Dept. of Electrical Engg.Government residential women's polytechnic Latur, 413532 (M.S.), India.

Abstract:

Autonomous 3D mobile robot mapping for environment monitoring has gained prominence due to its potential for eliminating human intervention. However, existing techniques often exhibit suboptimal performance in complex environments. To overcome this limitation, this paper introduces a novel framework known as "3DMRM-MP" (3-Dimensional Mobile Robot Mapping and Motion Planning using Deep Q-Learning-based Markov Decision Model Deep Neural Network). This framework relies on primary sensors for robot navigation. It involves pre-processing point clouds and grouping similar pixels to extract features, with an added enhancement step using the Gazelle Optimization Algorithm (GOA). Estimation of the robot's current posture is achieved through the Transformation matrix applied Single value decomposition Linear N-Point Camera Pose Estimation (TMSVDLCPE). Based on this estimated pose, the framework determines the desired view, captures images, and converts them into 3D formats. The robot's 3D images, speed, and current position serve as inputs to the DQMD-DNN, which efficiently plans the robot's next optimal move. Experimental results demonstrate that the proposed technique achieves significantly higher decision accuracy compared to existing approaches.

Keywords: Light Detection and Ranging (LIDAR); Inertial Measurement Unit (IMU); Brownian motion (BM); Bayes Distribution-Gazelle Optimization algorithm (BD-GOA); Deep Q-Learning-based Markov Decision Model Deep Neural Network (DQMD-DNN).

1. INTRODUCTION

Omnidirectional mobile robots offer several advantages, including strong mobility, precise positioning, and simple control (Wang et al., 2020). One of the fundamental challenges in mobile robotics is navigation, where the goal is to identify the optimal path between a starting point and an end target (Perminov et al., 2021). Over the past few decades, wheeled autonomous robots, in particular, have seen significant advancements (Xue et al., 2022). In environments where GPS signals are unreliable or absent, a Simultaneous Localization and Mapping (SLAM) system plays a critical role (Wang et al., 2022). Additionally, an exploration module that utilizes the generated map and the robot's localization information is necessary to determine the next best goal (Zhu et al., 2020).

It's worth noting that existing approaches have often overlooked the quality of mapping (Yang et al., 2023). Unlike previous path-planning methods, which generate fixed paths based on maximal obstacle clearance, little attention has been paid to establishing a unified system for integrating and analyzing various constraints affecting robot navigation (Jud et al., 2021). However, this field faces two major challenges: (1) designing effective image features to represent image information and (2) addressing potential failures due to changes in illumination, camera parameters, object movements, and environments lacking texture (Xu et al., 2021; Xue et al., 2020).

1.1. Problem Statement

Our work is motivated by the following factors:

- While LIDAR-based mapping yields accurate results, the quality of the mapping can be improved.
- Manual control is required for navigating unknown environments, leading to potential inaccuracies.
- Many existing algorithms have longer exploration times than desired.

1.2. Objectives

The objectives of our proposed technique are as follows:

- Enhance LIDAR-based mapping quality through efficient preprocessing.
- Develop robust decision-making systems capable of handling navigation in unfamiliar or challenging terrains without human intervention.
- Create efficient exploration strategies in path planning to optimize exploration time without sacrificing mapping accuracy and coverage.The remainder of this paper is organized as follows: Section 2 provides a survey of related work, Section 3 presents our proposed system, Section 4 discusses the outcomes, and Section 5 concludes the paper.

2. LITERATURE SURVEY

(Gobhinath et al., 2021) propounded SLAM and middleware concepts like open source (ROS). For the cost at which the presented mechanism was constructed, this distance was very higher. But, arrangement files were well-defined with setup limitations as well as startups so that the equipment was to be wielded. (Shin & Na, 2020) proffered an approach for displaying the elevation and temperature of a surveillance area in the form of a map by utilizing the Convolution Neural Network (CNN) algorithm. Outcomes displayed a better detection rate. Nevertheless, the reconstruction error for the primary anomaly was higher.(Diane et al., 2019) established multi-aspect mapping technology at the level of their semantic representation. The developed mapping subsystem's inputs were a series of Red Green Blue-Depth (RGB-D) streams. However, while using large sets, the time extends and creates unwanted classes.(Noh et al., 2020) introduced a system that could autonomously navigate an unstructured indoor environment, which avoided collision with static or else dynamic objects. Although it performed well in collision avoidance, still infer about object intentions was lower.(Huang et al., 2019) developed a technique of concurrent construction of 2D as well as 3D maps grounded on the mobile robot. For achieving the robot's pose, the particle filter approach was wielded, which had superior performance. Yet, the map resolution was very low.

(Tang et al., 2020) presented a 3D exploration system centered on the wavefront framework. The experimental results demonstrated a higher efficacy. But, under the people's direct control of the environment, mapping could not be completed by mobile robots.

(Eldemiry et al., 2022) propounded an exploration technique for concurrently optimizing exploration time by utilizing a lower-cost RGB-D camera. Owing to low computational cost and low exploration time, feature-centric RGB-D SLAM was wielded. Nevertheless, the online mapping quality was lower.

3. PROPOSED METHODOLOGY FOR 3D MOBILE ROBOT MAPPING AND PATH PLANNING

The proposed model efficiently identifies the obstacles in the robot's path and makes optimal decisions for navigation, such as choosing a new path or adjusting the robot's speed and direction. Figure 1 elucidates the proposed system architecture.

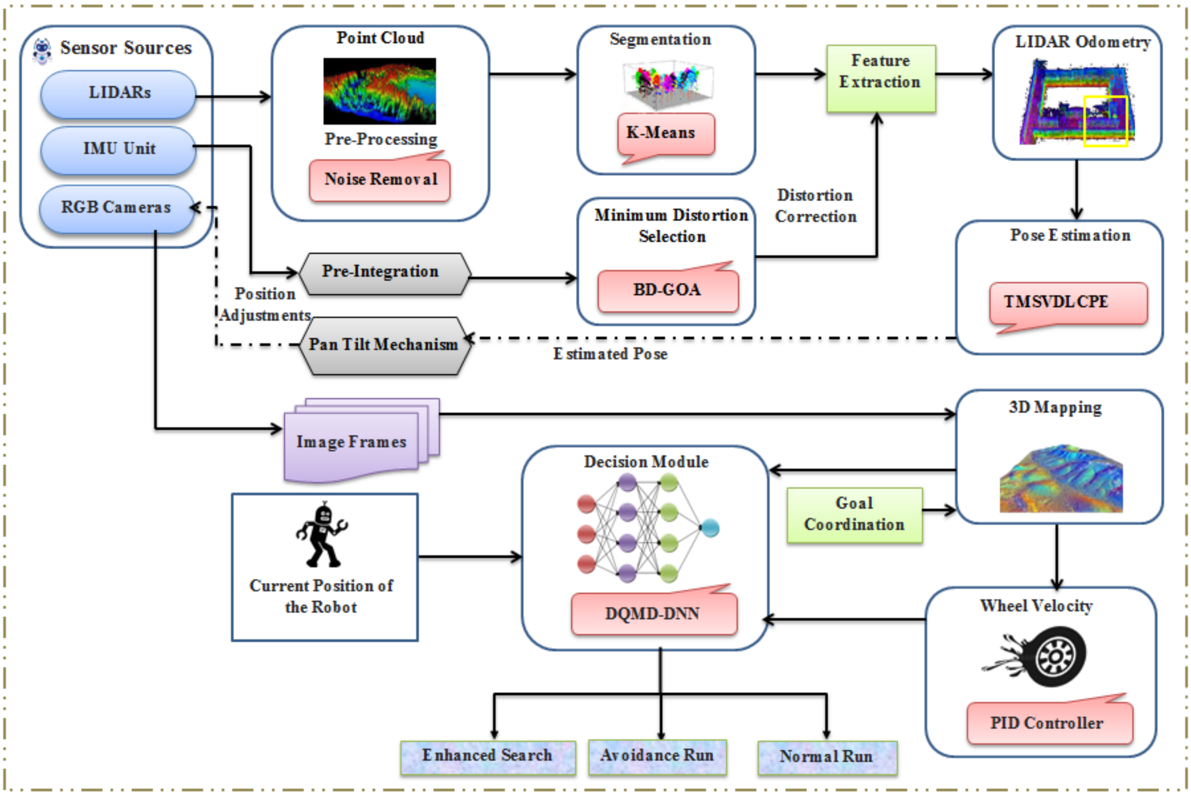


Figure 1: Architecture of the proposed framework

3.1. Sensor sources

LIDAR: LIDARs are utilized for mapping, localization, and navigation. It creates a point map (point cloud) of its surroundings.

Inertial Measurement Unit (IMU): It measures linear acceleration, angular velocity, and the magnetic field around the robot respectively.

RGB cameras: The images captured by the RGB cameras aid the robot identify objects and features in its surroundings.

3.2. Preprocessing

The point clouds from the LIDARs are preprocessed for removing noise and enhance the resulting point cloud's quality using a Gaussian Filter (GF). The GF is defined by,

$$G(a,b,c) = \left(\frac{1}{2\pi SD^3} \right) * Exp\left(-\left(\frac{a^2 + b^2 + c^2}{2SD^3} \right) \right) \tag{1}$$

Where, a, b and c are the pixel coordinates and SD exemplifies the standard deviation. The preprocessed point cloud is P_c.

3.3. Segmentation

Grounded on the pixel similarity, the pixels from P_c are grouped together using the K-Means algorithm, which is scalable to large datasets. The grouping of pixels aids the robot to understand the environment structure in a better way. The initial cluster centroid (Cn_i) is selected randomly. The pixels (Px_i), which have a minimum distance with (Cn_i), are assigned to the clusters. The clustering distance $(Dist)$ is computed by,

$$Dist_i = \sqrt{\sum_{i=1}^{n}(Cn_i - Px_i)^2}, \qquad Dist_i, i = 1,2,3,......,n \tag{2}$$

Here, i implies iteration.

3.4. Distortion correction

When a robot moves through its environment, the motion can cause distortions in the point cloud, which make it difficult to accurately extract features. Thus, to deal with this, pre-integrated IMU outputs are wielded that correct the distortions and enhance the feature extraction accuracy. The pre-integrated IMU outputs represent the expected change in orientation and position of the robot over a while. Therefore, the minimum distortions (M_d) are selected by BD-GOA. The GOA can avoid local optima and get a globally optimal solution. The GOA uses BM in the exploitation phase. However, the unequal step length in BM might lead to false convergence. To overcome this, Bayes Distribution (BD) is employed for the step length of BM. The steps of BD-GOA are defined further,

Primarily, the gazelle population (represents the position of the robot over a time $(p_{i,j})$) is initialized by,

$$p_{i,j} = r \times (Ub_j - Lb_j) + Lb_j \tag{3}$$

Where, r is a random number, Ub_j and Lb_j are upper and lower bound, correspondingly.

After that, the fitness Ft is computed grounded on the minimum distortion $Min(Ds)$, which is referred to as,

$$Ft = Min(Ds) \tag{4}$$

Exploitation: When the gazelles are stalked by predators while grazing, the gazelles move in BM, this process is defined by,

$$\vec{g}_{i+1} = \vec{g}_i + k \cdot \vec{R} * \vec{B}_R * (\vec{E}_i - \vec{B}_R * \vec{g}_i) \tag{5}$$

Where, $\vec{g}_{i+1}$ and $\vec{g}_i$ are the solution of the next iteration and current iteration, correspondingly, k symbolizes grazing speed, $\vec{E}_i$ implies elite, $\vec{R}$ is random numbers in [0, 1], and $\vec{B}_R$ is a BM value. In BM, the step length is determined by BD,

$$Pb(St|L) = Pb(L|St) * Pb(St) / Pb(L) \tag{6}$$

Where, $Pb(St)$ is the prior probability of step length, $Pb(L)$ is the total probability of overall step length.

Exploration: Once the predator is spotted, the gazelle runs, and the predator chases. The sudden change of direction u is,

$$\vec{g}_{i+1} = \vec{g}_i + K \cdot u \cdot \vec{R} * \vec{L}_R * (\vec{E}_i - \vec{L}_R * \vec{g}_i) \tag{7}$$

Where, K describes the top speed of gazelle and $\vec{L}_R$ signifies Lévy distributions. The behaviour of the predator chasing the gazelle is displayed as,

$$\vec{g}_{i+1} = \vec{g}_i + K \cdot u \cdot cf * \vec{B}_R * \left(\vec{E}_i - \vec{L}_R * \vec{g}_i\right) \tag{8}$$

Where, $cf = \left(1 - \frac{itr}{Max_{itr}}\right)^{\left(2\frac{itr}{Max_{itr}}\right)}$ signifies the controlling parameter. Grounded on (M_d), the point cloud is corrected S_C.

Input: Position of the robot over a time $(p_{i,j})$
Output: Minimum distortion (M_d)
Begin
Initialize the population $(p_{i,j})$
Compute the fitness value (Ft)
While $(i = 1 to\, i_{Max})$ do

Update the gazelle's position using,

$$\vec{g}_{i+1} = \vec{g}_i + k \cdot \vec{R} * \vec{B}_R * \left(\vec{E}_i - \vec{B}_R * \vec{g}_i\right)$$

Re-compute the fitness value (Ft)
If $(Ft == Satified)$
End the iteration
Else

Update the gazelle's position using,

$$\vec{g}_{i+1} = \vec{g}_i + K \cdot u \cdot \vec{R} * \vec{L}_R * \left(\vec{E}_i - \vec{L}_R * \vec{g}_i\right)$$

End if
$\vec{g}_{i+1} == Ft$
End while
Return (M_d)
End

3.5. Feature extraction

From S_C, important features (F) like line (detect walls, objects edge, and other straight or curved features in the environment), gradient (detect changes in elevation or slope in the environment), along with spatial features (track objects in the environment) are extracted.

3.6. LIDAR Odometry

Afterward, the LIDAR odometry (OT) is constructed using F, which estimates the robot's position and orientation. This is important for robot navigation and localization.

To construct (OT), (F) are matched between consecutive LIDAR scans. Once the matched features are identified, their relative motion is estimated by,

$$OT = Arg \min \sum \left(D(F_X, F_Y)\right)^2 \tag{9}$$

Where, D symbolizes Euclidean distance, F_X and F_Y are corresponding features in the point clouds.

3.7. Pose estimation

Grounded on the estimated motions (Es), the robot's current posture in an environment is determined using a TMSVDLCPE, which is computationally efficient. In LCPE, the Projection Matrix (PM) is wielded for estimating the camera pose. But, the PM can result in an incorrect estimation of the camera pose owing to degenerate cases. To overcome this issue, a Transformation Matrix (TM) is utilized. The steps are as follows:
Top of Form
Bottom of Form

Step 1: Normalize the odometry image points by,

$$c_1 = (c_1 - x_a) / fx_a \tag{10}$$

$$c_2 = (c_2 - x_b) / fx_b \tag{11}$$

Where, c_1 and c_2 are the normalized image coordinates, x_a and x_b are principal point coordinates, fx_a and fx_b are focal lengths.

Step 2: The measurement matrix $[M]$ is constructed from the normalized image points and 3D coordinates of the known points $[A],[B],$ and $[C]$.

$$[M] = \begin{bmatrix} [A] & [B] & [C] & 1 & 0 & 0 & 0 & -c_1[A] & -c_1[B] & -c_1[C] & -c_1 \\ 0 & 0 & 0 & 0 & [A] & [B] & [C] & -c_2[A] & -c_2[B] & -c_2[C] & -c_2 \end{bmatrix} \tag{12}$$

Step 3: The SVD of $[M]$ is computed, this yields left (L) and right (R) singular vectors (Sv). The SVD is,

$$M = L * Sv * R \tag{13}$$

Step 4: After that, TM is computed by,

$$Tm = \mathrm{Im} * Rm \tag{14}$$

Step 5: Lastly, the camera pose is recovered by decomposing TM into the camera extrinsic matrix (Em) and the intrinsic matrix (Im) by,

$$[Em, \mathrm{Im}] = rq(Tm(1:3)) \tag{15}$$

$$Tv = \mathrm{Im}^{-1} * Tm(:4) \tag{16}$$

Where, rq is decomposition factor, Tv is translation vector, $Tm(1:3)$ selects all rows and the first three columns of Tm, and $Tm(:4)$ selects all rows and the fourth column of Tm. Therefore, the estimated pose is defined by Ep.

3.8. Pan tilt mechanism

Centered on Ep, the Pan Tilt mechanism controls the movement of the RGB to capture a desired view of the scene. The captured images are further processed to extract frontier and area information, which provides valuable insights about the environment.

3.9. 3D mapping

To navigate to a destination point, the captured images are converted into 3D maps along with the goal coordinates. 3D mapping allows robots to better understand their environment by creating a detailed map in three dimensions. With a 3D map of the environment, robots can more easily navigate to a destination point by calculating the most efficient path and avoiding obstacles. Moreover, the 3D map can provide valuable information about the terrain, such as the height of objects and the depth of water, which can be utilized for making more informed decisions about how to move through the environment.

Top of Form

3.10. Wheel velocity

From 3D mapping, the information, such as the method of movement, track deviation from the landmark, and desired landmark are inputted to the Proportional-Integral-Derivative,
(PID) controller, which adjusts the wheel velocity to attain desired motion. The PID controller is expressed as,

$$T = G_P * E(t) + G_I * \int E(t)dt + G_D * de(t)/dt \tag{17}$$

Where, the control parameter is notated as T, proportional, integral, and derivative gains are exemplified as G_P, G_I and G_D, correspondingly, the error rate at the time t is specified as $E(t)$, the error over time's integral is signified as $\int E(t)dt$, and the derivative of the error at t is elucidated as $de(t)/dt$.

3.11. Decision making

The robot's current position, current velocity, and 3D maps are inputted to the DQMD-DNN, which efficiently makes decisions about the optimal action to take, namely normal run, enhanced search, and avoidance run.

Reinforcement learning is used by the DQMD-DNN to train the DNN by iteratively adjusting the Q-values grounded on feedback from the environment. To maximize the reward function, this work uses MD. The MD explores the complex environment and adapts to change the environment over time, allowing it to continue learning and updating the optimal rewards. Hence, the algorithm makes better decisions about how to navigate the environment over time. Figure 2 displays the DQMD-DNN architecture.

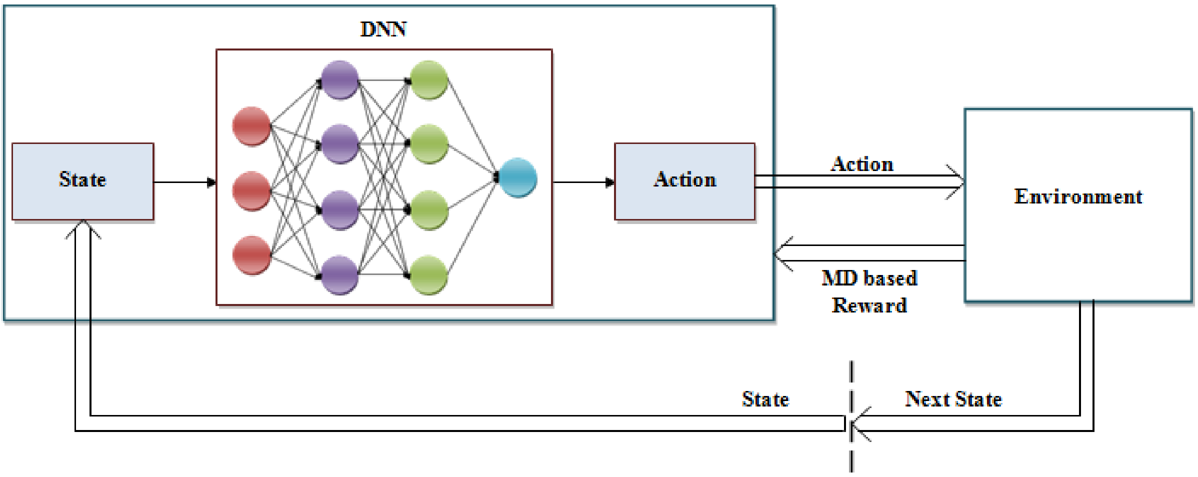


Figure 2: DQMD-DNN architecture

Afterward, the Q-values for each possible action are computed. The expected reward to take a specific action in the current state is represented by the Q-values.

$$Q(s,a) = Rw\pi(s,a) + \gamma * Max[Q(\bar{s},\bar{a})] \tag{18}$$

Where, the Q-value of the current state (s)-action pair (a) is notated as $Q(s,a)$, γ defines discount factor, $Max[Q(\bar{s},\bar{a})]$ is the maximum Q-value over all possible actions $\bar{a}$ in the next state $\bar{s}$, and $Rw\pi(s)$ is the reward obtained MD, this is given by,

$$Rw\pi(s) = ex\pi[Rw_{t+1} + \gamma Rw\pi\bar{s}] \tag{19}$$

Where, $ex\pi[Rw_{t+1} + \gamma Rw\pi\bar{s}]$ is the expected reward for the next state, and the reward received for taking action in the state (s) is represented as Rw_{t+1}.

Then, the best action is selected grounded on the Q-values by,

$$B^* = Arg\max - B(Q(s,a)) \tag{20}$$

Where, B^* is the best action, $Arg\max - B$ symbolizes the action, which maximizes the Q-value.

The Q-value is updated for B^* using the Q-learning update rule,

$$Q(s,a) = Q(s,a) + \beta(Or + \gamma MaxQ(\bar{s},\bar{a}) - Q(s,a)) \tag{21}$$

Where, the learning rate is exemplified as β, the observed reward is implied as Or, the discount factor for future rewards is notated as γ.

Lastly, the loss function (Ls) is determined by,

$$L = (Q(s,a) - y)^2 \tag{22}$$

Where, y is the target Q-value; the parameters are adjusted through back-propagation to minimize (Ls).

4. RESULTS AND DISCUSSION

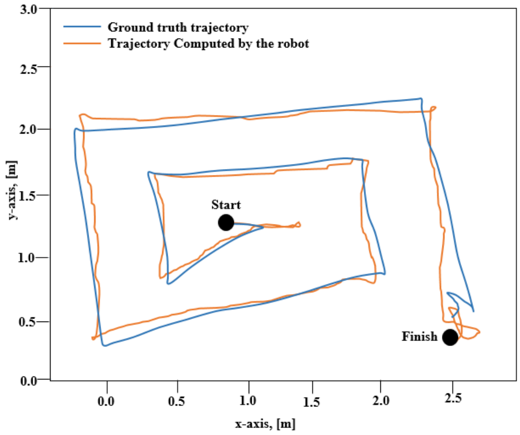
Here, the experiments conducted in the working platform of MATLAB are presented.

4.1. Dataset description

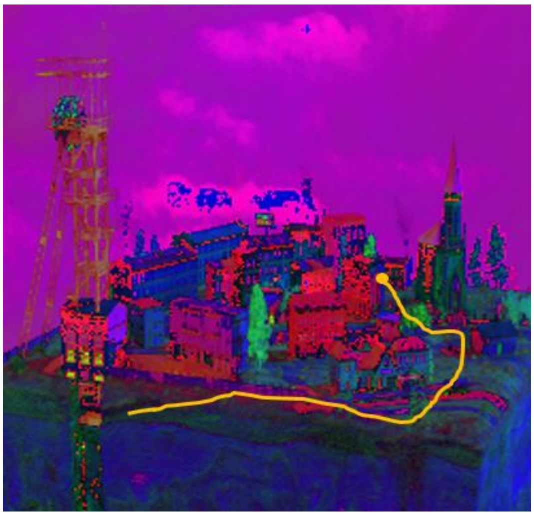
Toronto-3D, which is obtained by a Mobile Laser Scanning (MLS) system in Toronto, Canada for semantic segmentation, is a large-scale urban outdoor point cloud dataset. It approximately covers 1 km of road and comprises about 78.3 million points.

4.2 Performance Analysis

Here, the proposed 3DMRM-MP's performance is validated.



(a)



(b)

Figure 3 (a): Trajectory computation (b) Autonomous exploration of robot

The trajectory computed by the robot and the ground truth trajectory is compared in Figure3 (a), whereas the robot's autonomous exploration ability is displayed in Figure 3 (b). From the analysis, it is clear that the proposed system efficiently aids the robot in avoiding obstacles and passing safely in the exploration.

Table 1: Performance of the proposed 3DMRM-MP in terms of Frontier detection, Obstacle detection, and decision accuracy

Techniques	Performance metrics		
	Frontier etection	Obstacle detection	Decision accuracy
Proposed 3DMRM-MP	97.26	97.96	98.32

The proposed technique's performance is demonstrated in Table 1. The proposed 3DMRM-MP efficiently handles the distortions caused by the LIDARs. Furthermore, the proposed. 3DMRM-MP uses TMSVDLCPE for pose estimation and adjusts the RGB cameras as per the estimated results.

Owing to this, the proposed 3DMRM-MP achieves 97.26% of frontier detection, 97.96% of Obstacle detection, and 98.32% of decision accuracy.

Table 2: Performance of the proposed 3DMRM-MP in terms of error rates

Techniques	Performance metrics		
	RMSE	Localisation error	Loss value
Proposed DQMD-DNN	1.76	2.36	1.68
DNN	6.78	7.34	4.87
RNN	9.38	10.58	6.74
LSTM	13.06	15.69	9.38

The proposed DQMD-DNN's performance is demonstrated in Table 2. The proposed DQMD-DNN is efficiently learned by the Deep Q-Learning method. Further, better rewards are selected by employing the MD technique. Hence, the RMSE, Localisation error, and Loss value of the proposed DQMD-DNN are 1.76%, 2.36%, and 1.68%, correspondingly. Therefore, the proposed DQMD-DNN is capable to make decisions with a limited error rate.

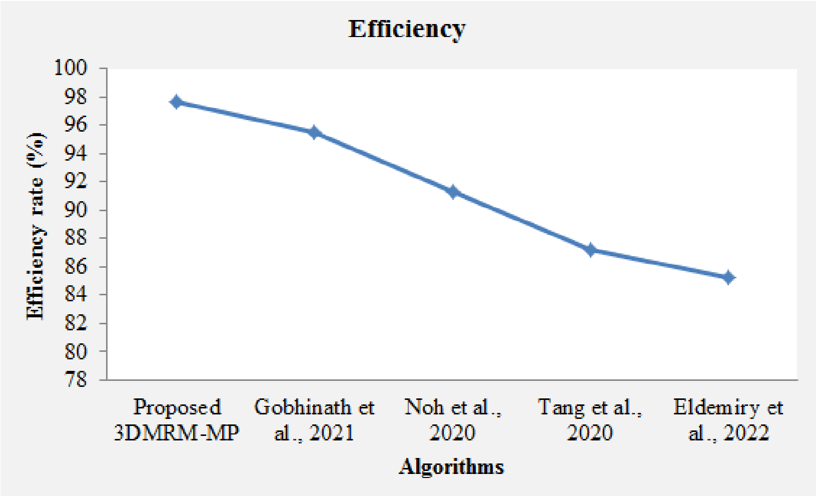


Figure 4: Efficiency comparison

Figure 4 displays the efficiency of the proposed 3DMRM-MP and the conventional mechanisms. The proposed 3DMRM-MP efficiently corrects the distortion rates and maintains the RGB cameras to capture the desired location. As a result of this, the proposed technique's navigation efficiency is higher than the prevailing research works.

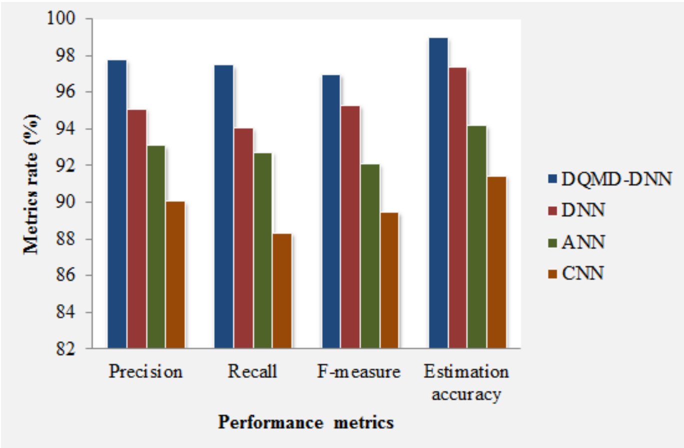


Figure 5: Performance comparison

The proposed 3DMRM-MP and the conventional techniques' performance are exhibited in Figure 5. The proposed 3DMRM-MP efficiently selects the reward. Thus, the proposed 3DMRM-MP achieves precision, recall, F-measure, and Estimation accuracy of 97.8%, 97.5%, 97%, and 99%, correspondingly, whereas the existing techniques obtain lower performance rates. Hence, the proposed E-RS-GRU classifies the attacked and non-attacked data efficiently.

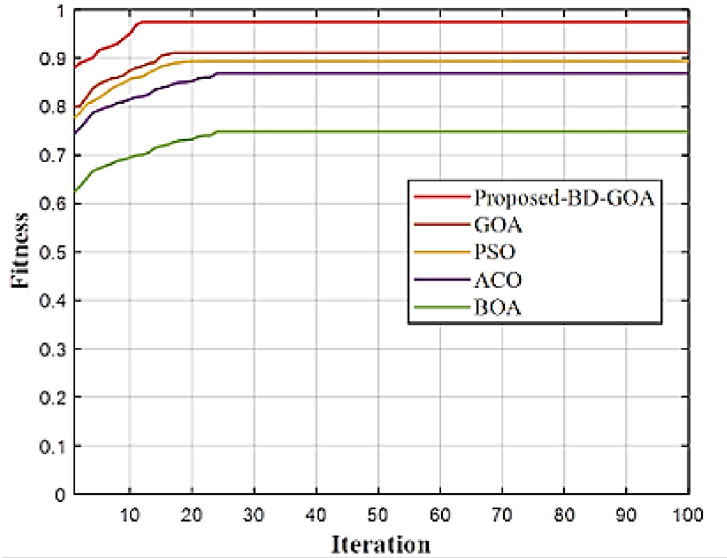


Figure 6: Fitness vs. Iteration Comparison

Figure 6 compares the proposed BD-GOA's optimization ability with Particle Swarm Optimization (PSO), Butterfly Optimization Algorithm (BOA), GOA, along with Ant Colony Optimization (ACO). The proposed BD-GOA improves the exploration phase with the BD technique. Thus, the proposed BD-GOA renders the optimum outcome with a minimum number of iterations; however, to attain convergence, more iterations are required by the conventional mechanisms.

5. CONCLUSION

This paper proposed 3-dimensional mobile robot mapping and motion planning using DQMD-DNN. The system undergoes several operations like preprocessing, segmentation, Distortion correction, Feature extraction, Odometry creation, 3D mapping, and Decision making. Afterward, the experimental assessment is conducted, where the proposed system's performance, as well as comparative evaluation, is executed for validating the technique's efficacy. The presented framework could handle several uncertainties and renders more promising outcomes. For the assessment, the point cloud LIDAR (Toronto 3D) dataset is wielded, where the proposed system attains 98.32% of decision accuracy. This work mainly concentrated on 3D mobile robot mapping in an unknown environment with autonomous control; however, there are some undefined problems in real-time environments like sudden climatic changes, unpredictable holes in the ground, et cetera. In the future, the work could be focused to make decisions grounded on these sudden undefined changes also.

Top of Form

REFERENCES

- Diane, S. A. K., Lesiv, E. A., Pesheva, I. A., & Neschetnaya, A. Y. (2019). Multi aspect environment mapping with a group of mobile robots. Proceedings of the 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, IEEE, 478-482. https://doi.org/10.1109/EIConRus.2019.8656876
- Eldemiry, A., Zou, Y., Li, Y., Wen, C. Y., & Chen, W. (2022). Autonomous exploration of unknown indoor environments for high- quality mapping using feature based RGB- D SLAM. Sensors, 22(14), 1-16. https://doi.org/10.3390/s22145117
- Gobhinath, S., Anandapoorani, K., Anitha, K., Sri, D. D., & Divyadharshini, R. (2021). Simultaneous localization and mapping [SLAM] of robotic operating system for mobile robots.

7th International Conference on Advanced Computing and Communication Systems, 1, 577-580. https://doi.org/10.1109/ICACCS51430.2021.9441758

- Huang, S., Li, C., Cai, Z., Zhu, G., Yao, L., & Fan, Z. (2019). Synchronized 2D SLAM and 3D mapping based on three wheels omni-directional mobile robot. Proceedings of the 9th IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, 1177-1181. https://doi.org/10.1109/CYBER46603.2019.9066733
- Jud, D., Hurkxkens, I., Girot, C., & Hutter, M. (2021). Robotic embankment. Construction Robotics, 5(2), 101-113. https://doi.org/10.1007/s41693-021-00061-0
- Noh, S., Park, J., & Park, J. (2020). Autonomous mobile robot navigation in indoor environments: Mapping, localization, and planning. International Conference on Information and Communication Technology Convergence, 908-913. https://doi.org/10.1109/ICTC49870.2020.9289333
- Perminov, S., Mikhailovskiy, N., Sedunin, A., Okunevich, I., Kalinov, I., Kurenkov, M., & Tsetserukou, D. (2021). UltraBot: Autonomous mobile robot for indoor UV-C disinfection. IEEE 17th International Conference on Automation Science and Engineering, 2147-2152. https://doi.org/10.1109/CASE49439.2021.9551413
- Qi, Y., Wang, R., He, B., Lu, F., & Xu, Y. (2022). Compact and efficient topological mapping for large-scale environment with pruned voronoi diagram. Drones, 6(7), 1-12. https://doi.org/10.3390/drones6070183
- Shin, H. C., & Na, K. (2020). Anomaly detection using elevation and thermal map for security robot. International Conference on Information and Communication Technology Convergence, 1760-1762. https://doi.org/10.1109/ICTC49870.2020.9289470
- Tang, C., Liang, Y., Yu, S., Sun, R., & Zheng, J. (2020). Autonomous 3D exploration of indoor environment based on wavefront algorithm. IEEE International Conference on Networking, Sensing and Control, 1-6. https://doi.org/10.1109/icnsc48988.2020.9238092
- Wang, C., Ma, H., Chen, W., Liu, L., & Meng, M. Q. H. (2020). Efficient autonomous exploration with incrementally built topological map in 3-D environments. IEEE Transactions on Instrumentation and Measurement, 69(12), 9853-9865. https://doi.org/10.1109/TIM.2020.3001816
- Wang, W., Jiang, L., Lin, S., Fang, H., & Meng, Q. (2022). Imitation learning based decision making for autonomous vehicle control at traffic roundabouts. Multimedia Tools and Applications, 81(28), 39873-39889. https://doi.org/10.1007/s11042-022-12300-9
- Xu, X., Wang, C., Wang, Y., & Xiong, R. (2021). HiTMap: A hierarchical topological map representation for navigation in unknown environments. arXiv preprint. http://arxiv.org/abs/2109.09293
- Xue, G., Li, R., Liu, S., & Wei, J. (2022). Research on underground coal mine map construction method based on LeGO-LOAM improved algorithm. Energies, 15(17), 1-17. https://doi.org/10.3390/en15176256
- Xue, W., Ying, R., Gong, Z., Miao, R., Wen, F., & Liu, P. (2020). SLAM based topological mapping and navigation. IEEE/ION Position, Location and Navigation Symposium, 1336-1341. https://doi.org/10.1109/PLANS46316.2020.9110190
- Yang, X., Lin, X., Yao, W., Ma, H., Zheng, J., & Ma, B. (2023). A robust LiDAR SLAM method for underground coal mine robot with degenerated scene compensation. Remote Sensing, 15(1), 1-19. https://doi.org/10.3390/rs15010186
- Zhu, J., Gehrung, J., Huang, R., Borgmann, B., Sun, Z., Hoegner, L., Hebel, M., Xu, Y., & Stilla, U. (2020). TUM-MLS-2016: An annotated mobile LiDAR dataset of the TUM city campus for semantic point cloud interpretation in urban areas. Remote Sensing, 12(11), 1-18. https://doi.org/10.3390/rs12111875