Unmanned Terrestrial Deep Stereo ConvNet Gofer Embedded with CNN Architecture

Sunil Karamchandani[1], Aayush S. Parikh [2], Arman A. Lalani [1], Dhruv M. Bhavsar [1], and Gaurang R. Kamat [1]
[1] Electronics and Telecommunication Department, DJSCOE, Vile-Parle (W), Mumbai–400056, Maharashtra, India
[2] Department of Mechanical and Manufacturing Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal –576104, India
Email: sunil.karamchandani@djsce.ac.in, aayushparikh01@gmail.com, armanlalani5@gmail.com, dhruvbhavsar912@gmail.com, kamat.gaurang@outlook.com

***Abstract*—The proposed project brings to life a robotic gofer with automatic speech recognition and obstacle avoidance mechanism using Deep Stereo ConvNet architecture. It is a cross between a chatbot and an unmanned driverless automobile. The physics, gesticulation, and motion of the designed gofer depends on the depth estimation, path planning, obstacle avoidance and navigation. The gofer chassis is designed using vehicle kinematics for the desired load. The bottom-up approach calculates the motor ratings, and develops the rotary encoder, wheel dimension and battery selection. The course of the gofer is deliberated using an orthogonally controlled differential drive. The microcontroller STM32F446 reads data from various sensors, communicates with ROS running on Raspberry Pi, and controls the two motors. The virtual environment for gofer is imported into a Gazebo world. A particle filtering algorithm is used for optimum tracking, CNN-based identification and control of the gofer. In the virtual environment, the gofer is tested on the realm of the school campus floor with Monte Carlo simulations and upgraded with neural network architecture using stereo images for object detection. The developed mobile application adds to the ease of gofer operation and control. The gofer contributes to the designed system as Merlin at the command of King Arthur.**

***Index Terms*—ROS, SLAM, Raspberry Pi, Optical Encoders, Gazebo, Blender, Flutter, CNN**

I. Introduction

Until the advent of COVID-19 in human civilization, Unmanned Ground Vehicles (UGV) were limited to performing convoluted tasks in exacting environments. They brazenly went in precincts with high radiation and temperature and were exclusive to military applications.

From the war front to our bedrooms, the UGVs have accomplished powerful breakthroughs with innovative intelligence being built into the system. Separating the teamster from his automobile has challenged innovators in a multitude of ways, including the task of obstacle avoidance, face detection, and speech emotions.

The system designed is a UGV with AI-based algorithms breathing life into it. A review by [1] on Human-Robot Interface (HRI) technologies suggests that it enhances consumer experience in the retail sector. Gofer is one such interface, but the human intervention is being kept to a bare minimum while providing service between the master controller and his subjects, tested in a campus arena. The contributions of the proposed work are abbreviated as follows:

The Unmanned vehicle, which works as a gofer, is designed on a Raspberry-pi using the forward vehicle kinematical equations. Its motion is PID controlled using an orthogonally controlled differential drive design to regulate the dc motor at the desired speed. ROS platform integrated on the pi computer assorted with Gazebo and RViz, which helps us in creating a virtual world environment and robotic model using SolidWorks. STM32F446 pools the data from sensors and establishes communication with the Raspberry-pi. The systematic design of the gofer: the hexagonal chassis design, wheel selection to generate maximum torque, motor drives for yaw determination, and battery load calculations along with current rating are presented in detail. Particle filtering helps the gofer navigate through the system and perform its duties as directed. This is done as a two-step process: the measurement update and time update. For the Robotic/URDF Model, a robot model in SolidWorks is created to generate a URDF file. A '.xacro' file version is imported into Ros (Gazebo and Rviz). The block diagram is as suggested in Fig. 1.

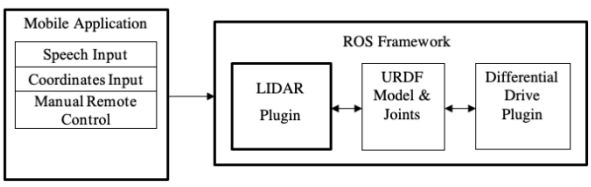

Figure 1. Block diagram of the proposed system.

Manuscript received August 1, 2022; revised October 13, 2022.

© 2022 Int. J. Mech. Eng. Rob. Res
doi: 10.18178/ijmerr.11.11.807-819

Using Blender, we created our university's 5th floor model of Department of Electronics & Telecommunication Engineering for the virtual environment. The generated *.blend* file is then converted into a *.world* file and imported into Ros (Gazebo and Rviz).

The simulation platform block diagram is suggested in Fig. 1. A mobile application is designed to control the entire system. Reproductions are performed on the Lidar in the virtual environment with a differential drive plug-in. The floor arena provides the vehicle a platform for path estimation, obstacle avoidance, object detection, face recognition, and finally, Automatic Speech Recognition.

SLAM is the holy grail for localization [2]. Over the years, sturdy SLAM algorithms have aided in robust tracking and obstacle avoidance. Odometry, Extended Kalman filter (EKF), Fast SLAM, this trio and more have proved helpful in varied terrains, maps, and environments [3]. Computational costs, large disparity maps, and problems in the particle filtering process (heart of the SLAM) have often hindered their application [4]. A review by authors [5] recommends that depth and other operational perfections have boosted the learning aptitude of the CNN and remain the best tools for vision-related tasks to date.

Here, the Deep ConvNet Convolution Neural Network comes to the rescue with its yin and yang approach of encoder and decoder [5]. This, of course, does away with the pose estimation backend of the SLAM. The gofer maneuvers through the maze of the objects and hands its control to the face recognition system, and finally, the speech recognition model takes over.

Once the networks are trained, the proposed method generates accurate detection in real-time applications effectively mounted over the Raspberry-pi module, making it easier for the gofer to commute. Merlin can be used as a delivery, cleaning robot, and many other applications. Perhaps in these times of the pandemic, its service can be extended as a sanitization bot by installing a UV lamp over it.

II. Literature Review

The term Simultaneous Localization and Mapping (SLAM) was first coined in a Mobile Robotics Survey paper presented in 1995 by H. Durrant-Whyte [6]. SLAM is a process by which a robot builds a map of its environment and surroundings and uses it to determine its current location. The platform's trajectory and the locations of the landmarks are predicted without prior knowledge of the location. When we use a single camera SLAM, the main drawback is that a single view cannot completely estimate its location. This drawback is addressed by the structure-from-motion (SFM) approach. It draws a conclusion by taking into account observations from multiple views. The main difference between SLAM and SFM is that SLAM requires online, real-time processing, whereas SFM primarily considers batch processing [7].

The autonomous driving of robots has evolved very quickly over a relatively short period. Idris and Arof [8] use Extended Kalman Filter (EKF) to estimate the robot position and motion information. It studies the Monocular SLAM EKF estimation process to identify the most time-consuming part of the estimation program. The authors in [9] use Adaptive Monte Carlo Localization (AMCL) algorithm and Bayesian algorithm for estimating the robot position and constructing the build map. It also uses Robot Operating System (ROS) for the development of the robot. The authors [10] use RGB-D cameras and compares selected Visual Odometry and Visual SLAM (V-SLAM) algorithms. Visual Odometry readings are subjective and therefore predisposed to errors.

Multi-Robot mapping using topological features [11] are used to increase the speed and mapping information between robots. Each robot explores different areas of same environment and they merge their data to form a global map using topological features to save the memory. This involves extensive hardware programming and data management capabilities.

The main goal of the authors in [12] is a mere performance comparison of existing SLAM algorithms Core SLAM, Gmapping, and Hector SLAM using ROS simulation. The authors have inferred that calculated Hausdorff Distance on three different maps and compared the results for different SLAM algorithms. The result found was HectorSLAM performed better than other two algorithms for two maps while Core SLAM was better in case of third map. The performance evaluation on mere three maps just does not provide sufficient confidence in the results.

Kinect sensor [13] is involved for mapping using open-source bundle G-Mapping and tele-operation technique to control the robot via RViz. Kinect sensor though it produces accurate results as compared to non-filtered laser, the algorithm is non-adaptive and might not produce results in all environments.

DIY Autonomous Mobile Robot is projected as a Chefbot in [14]. However, as a constraint of its application ground clearance of the robot should be greater than 3 cm.

Path planning algorithm using fuzzy logic [15] lays the path from source to destination without colliding with obstacles in between within a shortest path in static as well as dynamic environment. Analysis for various fuzzy rules is done for path planning of a mobile robot using various combinations of membership function. However, the execution of the membership function requires a very high computation time which proves hazardous in case of robot navigation.

The objects in the indoor environment [16] are being mapped by using a combination of portable RFID (Radio Frequency Identification) reader, UHF (Ultra High Frequency) passive tags and Inertial Navigation Sensors (INS) considering the indoor objects to be non-stationary in nature. INS however involve high maintenance cost and snag due to heat dissipation.

© 2022 Int. J. Mech. Eng. Rob. Res

RGB-D SLAM has been implemented in [17] which uses the vanishing point and door plates for navigating in a corridor environment. However, the estimation of the vanishing point requires implementation of extensive image processing algorithms.

Dijkstra algorithm has been used to analysis the shortest path to be used by the robot and obstacle avoidance [18]. Camera has been mounted on a domestic robot to acquire to display the outside scenery via Bluetooth to the mobile that will be used by the patients who have motion problems and paralysis in order to enjoy the virtual trip performed by the mobile robot. Bluetooth obviously will drain the power, moreover Dijkstra performs a blind search entailing annihilation of multiple resources.

Case study of a robotic excavator has been done [19] which has adopted Real-Time Control System (RCS) architecture for intelligence system and POMDP (Partial Observable Markov Decision Processes) has been used to form the basis of making decisions in uncertain situations. POMDP are probabilistic models which might work only for specific applications, one of them being excavation.

The set-up as suggested in [20] has integrated laser scanners such as Light Detection and Ranging (LiDAR), 3D cameras, accelerometers which make it quite an expensive affair.

Logging as a service (LAAS) architecture for autonomous systems, based on the component based design approach is implemented in [21]. There are three levels in this architecture functional level for image segmentation and motion control, followed by the decision level and then finally to the execution level which are based on multiple rules and constraints. Not only are the three functionalities required to work in tandem but optimization algorithms need to be executed for taking care of the constraints.

The authors in [22] take on contemporary robotics and automation and the way it reshapes the existing urban digital infrastructure. The manuscript provides us with a psychoanalytical view, how our gofer would be required to adjust in urban setting.

An inspection of autonomous vehicle in an urban environment [23] focuses on localization and tackles the complexities vehicle tracking. The behavioral pattern of the gofer can be useful in this study. The contemporary research [24] justifies the use of infrared-visual odometry using edge points as a substitute for infrared cameras thus escalating the costs, complex image processing and very objective inferences due to its visual odometry.

The importance of safety and reliability of robots in real-life scenarios is dealt with in [25] ensuring a steady growth and acceptance of robots for mission critical applications along with regulations for commercial robots which helps us to take the necessary precautions in our proposed design.

The manuscript in Section III details the system design from the intrinsic components to its Monte Carlo software implementation for navigation and obstacle detection. Section IV outlines the results, which are discussed in detail. Conclusion in Section V followed by future capabilities to be incorporated in the Gofer functionalities in Section VI.

III. Proposed System Design

The proposed system design is structured as follows: Part-A discusses the kinematics model of the UGV. Part-B mirrors the load calculations, motor selection along with the chassis design, while part-C is the Monte-Carlo Algorithm integrated into the system.

A. Kinematics

A differential drive robot has two wheels actuated independently using two motors. The robot will move in a straight line (forward or backward) if both the wheels rotate in the same direction. The orientation (yaw) of the robot is controlled by turning the motors at different speeds. So, if the left wheel rotates faster than the right wheel, the robot will rotate in an anti-clockwise direction and vice-versa.

As shown in Fig. 2,

ω = Angular velocity of the bot

v_r = Linear velocity of the right wheel

v_l = Linear velocity of the left wheel

r = Wheel radius

L = Distance between the two wheels

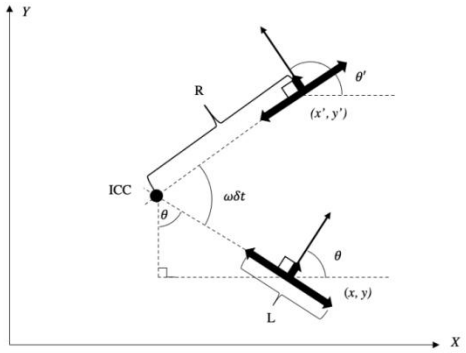

Figure 2. Schematic diagram of the two wheels bot in motion.

As,

$$v_l = r_l\omega = \left(R - \frac{L}{2}\right) \times \omega \quad (1)$$

$$v_l = R\omega - \frac{\omega L}{2} \quad (2)$$

$$v_r + v_l = 2R\omega \quad (3)$$

Therefore,

$$R = \left(\frac{v_r + v_l}{v_r - v_l}\right) \times \frac{L}{2} \quad (4)$$

Similarly,

$$v_r = r_r\omega = \left(R + \frac{L}{2}\right)\omega \quad (5)$$

$$v_r - v_l = \omega \quad (6)$$

© 2022 Int. J. Mech. Eng. Rob. Res

$$\omega = \left(\frac{v_r - v_l}{L}\right) \tag{7}$$

Suppose the bot moves with angular velocity ω for δt seconds.

$$\theta' = \omega\delta t + \theta \tag{8}$$

$$ICC = \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix} \tag{9}$$

$$= \begin{bmatrix} x - R\cos\left(\frac{\pi}{2} - \theta\right) \\ y + R\sin\left(\frac{\pi}{2} - \theta\right) \end{bmatrix} \tag{9a}$$

$$= \begin{bmatrix} x - R\sin\theta \\ y + Rcos\theta \end{bmatrix} \tag{9b}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) \\ \sin(\omega\delta t) & \cos(\omega\delta t) \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \end{bmatrix} \tag{10}$$

$$= \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) \\ \sin(\omega\delta t) & \cos(\omega\delta t) \end{bmatrix} \begin{bmatrix} x - (x - R\sin\theta) \\ y - (y - R\cos\theta) \end{bmatrix} + \begin{bmatrix} x - R\sin\theta \\ y - R\cos\theta \end{bmatrix} \tag{11}$$

$$= \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) \\ \sin(\omega\delta t) & \cos(\omega\delta t) \end{bmatrix} \begin{bmatrix} (R\sin\theta) \\ -(R\cos\theta) \end{bmatrix} + \begin{bmatrix} x - R\sin\theta \\ y - R\cos\theta \end{bmatrix} \tag{12}$$

$$= \begin{bmatrix} R\sin\theta\cos(\omega\delta t) + R\cos\theta\sin(\omega\delta t) \\ R\sin(\omega\delta t)\sin\theta - Rcos\theta\cos(\omega\delta t) \end{bmatrix} + \begin{bmatrix} x - R\sin\theta \\ y + R\cos\theta \end{bmatrix} \tag{13}$$

Therefore,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} R\sin(\theta + \omega\delta t) \\ -Rcos(\theta + \omega\delta t) \end{bmatrix} + \begin{bmatrix} x - R\sin\theta \\ y + R\cos\theta \end{bmatrix} \tag{14}$$

Generalizing for 3 dimensional,

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} \cos(\omega\delta t) & -\sin(\omega\delta t) & 0 \\ \sin(\omega\delta t) & \cos(\omega\delta t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - ICC_x \\ y - ICC_y \\ \theta \end{bmatrix} + \begin{bmatrix} ICC_x \\ ICC_y \\ \omega\delta t \end{bmatrix} \tag{15}$$

$$= \begin{bmatrix} R\sin(\omega\delta t + \theta') \\ -R\cos(\omega\delta t + \theta) \\ \theta \end{bmatrix} + \begin{bmatrix} x - R\sin\theta \\ y + R\cos\theta \\ \omega\delta t \end{bmatrix} \tag{16}$$

$$= \begin{bmatrix} -R\sin\theta + R\sin(\omega\delta t + \theta) \\ R\cos\theta - R\cos(\omega\delta t + \theta) \\ \omega\delta t \end{bmatrix} + \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{17}$$

R and ω are related to v_r and v_l

In 1 revolution,

C ticks $\equiv 2\pi r$ units (linear distance)

x ticks $\equiv \frac{2\pi rx}{C}$ units

Therefore,

$$velocity = \left(\frac{2\pi rx}{C}\right)/\delta t \tag{18}$$

B. Parameter Design

The UGV design is a simple prototype with a multitude of everyday applications. Battery calculations abetted with motor and wheel selection are detailed in the following section.

1) Battery Load Calculation

The equation (19) provides a general rule for the power calculation of the designed system

Calculated Load = (Current consumption * No. of Optical encoders) + (Current consumption * No of Arduino Due) + (Current consumption * No of Raspberry Pi) + (Current consumption * No of IMU Sensor) + (Current consumption * No. of Cytron Motor Driver) + (Current consumption * No of Ultrasonic Sensor) (19)

In our proposed design

No. of Optical encoders = 2

No. of Arduino = 1

No. of Raspberry Pi = 1

No. of IMU Sensors = 1

No. of Motor drivers = 1

No. of Ultrasonic sensors = 4

Calculated Load = 2.11 A

For working hours' h',

Battery Capacity Required = Load * Hours = 2.11 *h

2.11h Ah = (2.11h) mAh

Therefore, Battery Capacity Chosen = (2.11h) mAh

2) Wheel Selection

The larger the wheel radius, the greater is the torque. Also, a wider wheel will be required compared to our robot dimensions. Among the commonly available wheels in the market, the most optimum wheel was chosen with Diameter = 100mm and Width = 45mm.

3) Motor Selection

Desired speed = 1 m/s

w = v/r

w = 1/0.05 = 20 rad/s

1 round/sec == 6.28 rad/sec

Therefore 20 rads/sec = 3.18 rounds/sec

That is 3.18 rounds per sec or 191 rounds per minute (191RPM)

Weight of the Bot (W_1) = 10 kg

Payload (W_2) = 2 kg

Total Weight (W) = $W_1 + W_2$ = 12 kg ~ 120N

Coefficient of friction (μ) = 0.6

© 2022 Int. J. Mech. Eng. Rob. Res

The maximum torque is required when the robot starts moving as it must overcome static friction.

This maximum torque can be obtained as follows –

$$\mu * N * r - T = 0 \quad (20)$$

where,

μ is the coefficient of friction,

N is the average weight acting on each wheel,

r is the radius of wheels, and T is the torque.

$N = W/2$ (in the robot, actuation is only for two wheels, so we are taking $W/2$ for computing the maximum torque).

Therefore, we get $0.6 * (120/2) * 0.05 - T = 0$

Hence, $T = 1.8$ *N-m* or 18 Kg-cm

Considering safety margin, motors with higher specs of 21 kg-cm and 200 RPM is chosen.

Fig. 3 shows the sensors used in the robot and Fig. 4 shows the assembled circuit of the robot.

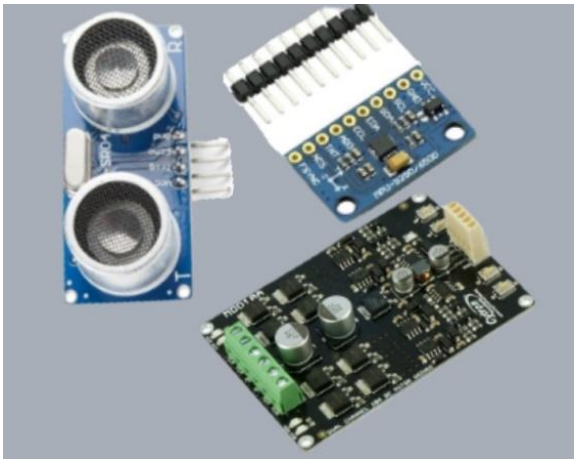

Figure 3. Sensors used in the robot.

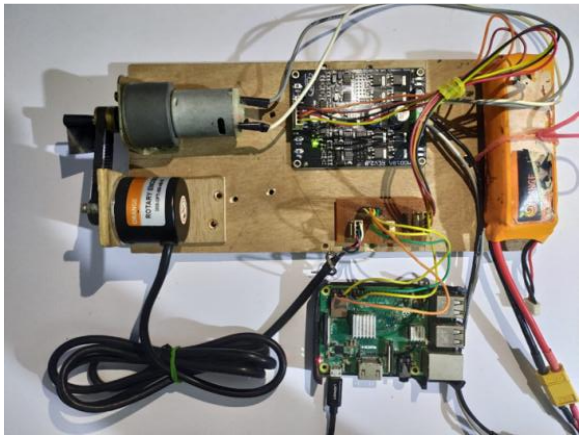

Figure 4. Hardware Setup for Motor Control

4) Chassis Design

The base of the robot is hexagon in shape. This shape was chosen as it is easy to manufacture when compared to shapes like circles or any other smooth surface. The edges in the hexagon provide an additional advantage of coupling multiple robots when needed to increase the payload.

The robot is divided into multiple stages. The lowest level or the base level has the Motors, Motor Driver, and encoders mounted on it. The second stage contains the micro-controller, Raspberry Pi, other sensors, actuators, and PCB.

The payload is placed on the topmost stage. There's the flexibility to add one more level if required. The stages are mounted on top of each other using studs.

C. Monte–Carlo Algorithm

1) Particle Filtering in Monte-Carlo Algorithm

i) Initially, the robot does not know where it is located. It estimates its pose by solving global localisation problem as shown in Fig. 5.

Figure 5. Robot trying to solve a localization problem

ii) The robot has a broad range of sensors to detect objects and walls to determine where it is located relative to its surroundings as shown in Fig. 6.

iii) The current robot pose is determined by x coordinate, y coordinate and orientation (θ).

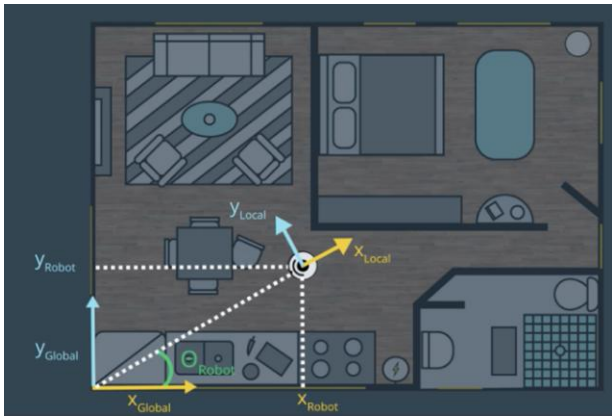

Figure 6. Robot determining current pose

iv) As per the MCL algorithm, particles are equally and uniformly spread across the simulated map. As shown in Fig. 7, each particle is defined by x coordinate, y coordinate, orientation (θ) and weight.

© 2022 Int. J. Mech. Eng. Rob. Res

Figure 7. Uniformly Distributed Particles as per MCL

v) The larger the weight, the larger the particle's probability of surviving. This means a more significant probability of the robot moving in that particle's direction, which can be seen in Fig.8.

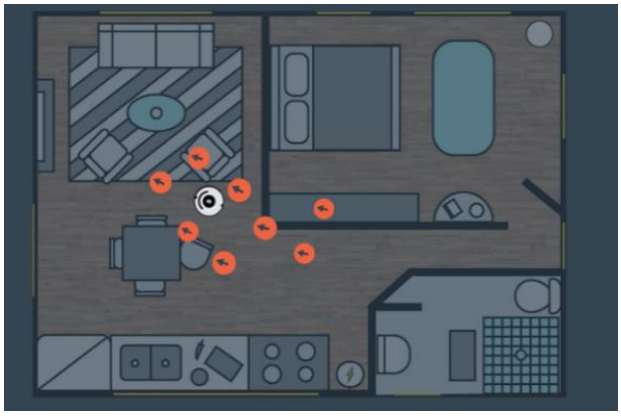

Figure 8. Robot moving towards larger weighted particles after few iterations.

vi) The algorithm detects the larger weight particles to reach the goal every iteration to estimate the robot's pose, like in Fig. 9.

This function keeps iterating and updating the robot's position by receiving the data from the onboard sensors until it reaches the final position, as seen in Fig. 10. and Fig. 11 shows the algorithm implemented using Rviz.

Figure 9. Robot moving towards larger weighted particles

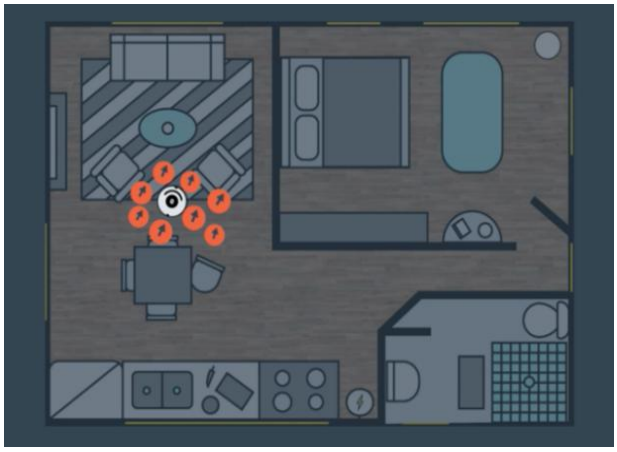

Figure 10. Robot reaches the destination after the final iteration.

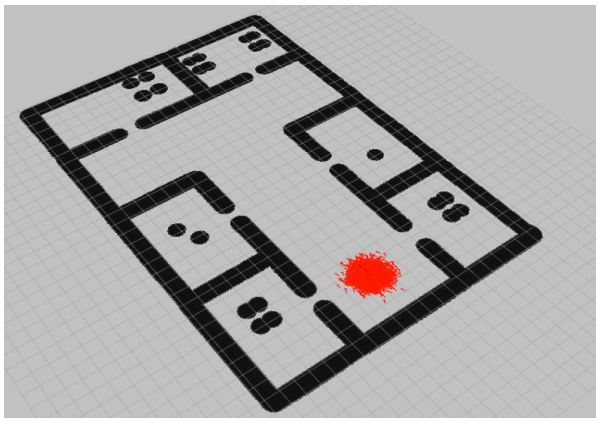

Figure 11. MCL algorithm implemented in the simulation when studied using RViz.

TABLE I. PSEUDOCODE OF MONTE-CARLO ALGORITHM

Pseudocode of Monte-Carlo Algorithm

function MCL (X_{t-1}, u_t, z_t)*:*

$\bar{X}_t = X_t = \phi$
for m=1 to M:
  $x_t^{[m]}$ = *motion_update* (u_t, $x_{t-1}^{[m]}$)
  $w_t^{[m]}$ = *sensor_update* (z_t,$x_t^{[m]}$)
  $\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
end for
for m=1 to M:
  draw $x_t^{[m]}$ *from* $\bar{X}_t$ *with probability* $\alpha\ w_t^{[m]}$
  $X_t = X_t + x_t^{[m]}$
end for
return X_t

Table I shows the pseudocode of the Algorithm used.

Fig. 12 shows how the robot looks in action when the algorithm is implemented in the simulation according to RViz.

© 2022 Int. J. Mech. Eng. Rob. Res

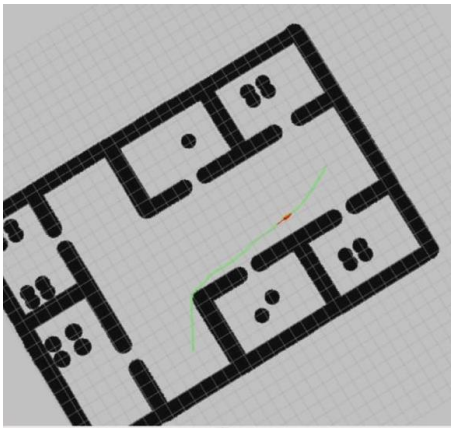

Figure 12. The green line represents the path which the algorithm has estimated to reach the given destination.

In Fig. 13, the weighted particles are drawn randomly and uniformly over the entire pose space.

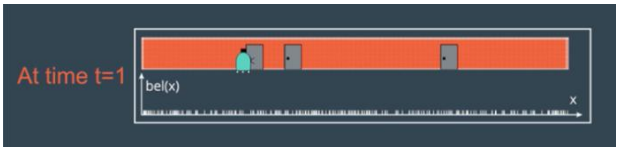

Figure 13. Position update at time t=1.

The taken measurement has been updated and priority weights have been assigned to each particle as seen in Fig. 14.

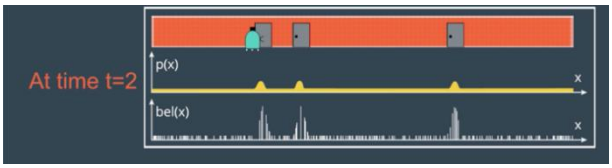

Figure 14. Position update at time t=2

In Fig. 15, The robot's motion has been updated and a new particle set with uniform weights and high number of particles around the three most likely places is obtained in resampling.

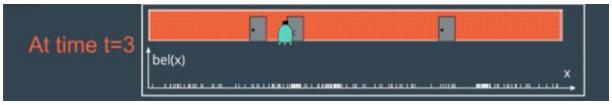

Figure 15. Position update at time t=3.

The new measurement assigns non-uniform weight to the particle set in Fig. 16.

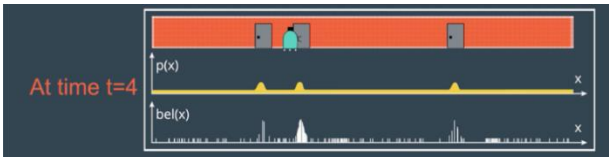

Figure 16. Position update at time t=4.

The new motion has been updated and a new resampling step is about to start as shown in Fig. 17.

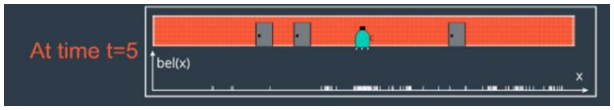

Figure 17. Position update at time t=5.

Fig. 18 shows the complete block diagram of the hardware system.

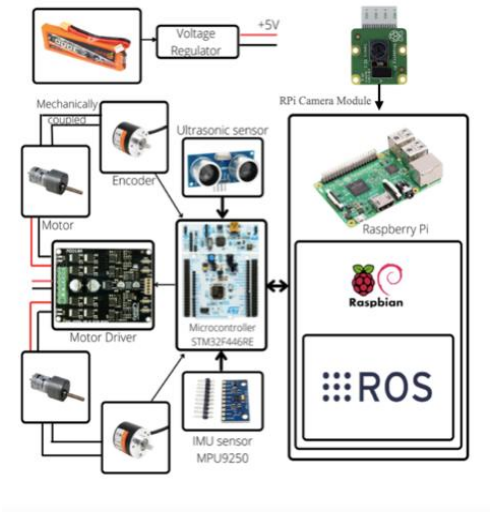

Figure 18. Block Diagram of the Hardware System Design.

D. *Integration of Gofer with AI Ensemble*

Strangely, the proposed gofer is trained more as Watson than Sherlock, a machine more innate than rational. Dealing now with bots under the influence of artificial intelligence, Bui et.al. utilize the power of CNN to predict all possible directions from images to decide the robot navigation [26]. The authors [27] use ConvNets to get the vanishing point to trace the path of the unmanned vehicle. As a precursor, the ConvNets use the features extracted from a Gabor filter, thus there is a duplicity in the feature extraction, foremost by the Gabor filter followed by those in the ConvNet architecture. If the road is a curved road, then a fuzzy network is used.

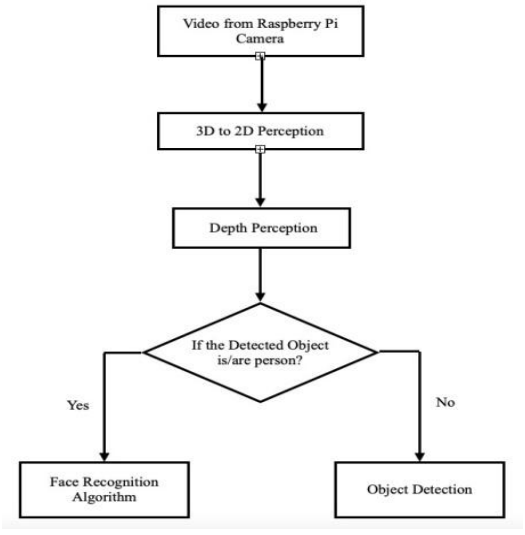

Figure 19. Logic diagram of decision-making process of the gofer.

© 2022 Int. J. Mech. Eng. Rob. Res

The proposed gofer is trained on any type of terrain with the Depth ConvNet SLAM localization finding its way out of maize of obstacles. What Sherlock would have done was imitate the human eye, locate the object distance using stereo image with rectified epi-polar lines. Captures of the same scene from different belvederes and estimate the distance of the objects and would have called it the depth of the scene.

Our schema uses the Watson way and develops a symbiotic machine architecture, a dual machine for depth estimation and a Convolution Neural network (CNN) for object detection. The former involves information about objects' distance from a vantage point. Fig. 19 details the journey of the gofer along with its integrated AI ensemble as the camera captures its voyage from the source carrying with it the speech information to be delivered to the mapped individual. 2D perception of the video is captured and fed to a neural network for object localization. The gofer searches for its target, evading obstacles, identifying the objects, and looking out for its target.

Disparity Estimation: The architecture proposed by Ibrahem [28] et al. for depth estimation generates a blocking effect in the projected depth. The authors employ a simple encoder architecture without the need for a decoder. The upsampling process introduced in the encoder generates the blocking effect.

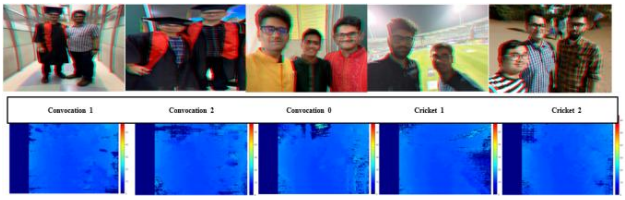

Figure 20. Depth Perception on sample 2D images

Lin [29] et al. propose a hybrid architecture of CNNs, one part extracting distinct features for depth estimation and the other finding universal features for both estimation and segmentation. The latter is deemed unnecessary, providing a bit of burden on the system design. Fig. 20 shows the depth perception carried out on sample 2D images.

The authors [Liu] [30] articulate depth estimation as a conditional random field (CRF) learning problem. So there is a graphical model unified with a CNN, which calculates the depth of an image by solving the maximum-a-posteriori (MAP) or point estimators. A deep learning neural network detects the disparity between the objects on the background, which considers the stereo matching between the images. In the Deeper Stereo ConvNet [2], input stays stable, but an additional convolution and deconvolution layer modifies the design. It can display distance values of the objects in the real world acquired from the Kinect sensor incorporated during the simulation. Fig. 20 displays the disparity map of the images acquired from the camera after its 2D perception from the captured video. Here the gofer takes a decision; if the object detected is a human, then a face recognition algorithm takes over for identification; else, the navigation system takes over to overcome the obstacle detection.

Table II shows the estimated depth values of the obstacles detected in the 2D images.

TABLE II. DEPTH ESTIMATION OF 2D PERCEPTED IMAGES

2D Percepted Images	SLAM	Proposed Deep Stereo ConvNet
Convocation_1	2.75	3.35
Convocation_0	0.78	1
Convocation_2	1.37	2.14
Cricket_1	2.05	2.65
Cricket_2	2.38	2.97

Face Recognition Gofer is employed on the school campus floor. The database of the faculty and the current batch of students are created, with six frontal and side faces each. The original dataset is converted into a large, augmented dataset by several transformations of the face images (120 x 6 = 720 images).

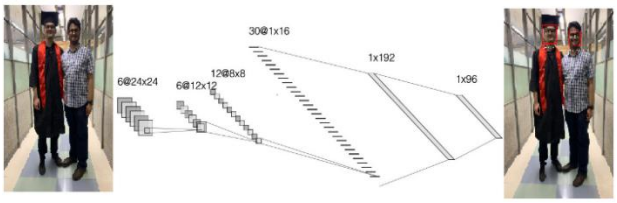

Figure 21. Facial Recognition using the proposed CNN.

The authors Cuevas [31] identify hydromorphological features in river images, with a Histogram of gradients provided as features to an SVM classifier. In addition, a Viola-Jones algorithm is cascaded to recognize the aforementioned features. Our proposed model, shown in Fig. 21, uses a CNN for face recognition which embraces the feature selection and identifier with sufficiently higher accuracy and lesser computations.

TABLE III. PERFORMANCE OF PROPOSED FACE RECOGNITION CNN

	Precision	Recall	F-Score
Predicted No	0.95	0.94	0.95
Predicted Yes	0.85	0.92	0.89
accuracy	0.94		
macro avg	0.90	0.93	0.92
weighted avg	0.91	0.91	0.91
Accuracy	93.73333		
Kappa Coefficient	0.96		

© 2022 Int. J. Mech. Eng. Rob. Res

The 24x24 images are max pooled to downsample the original images precursor to the feature extraction layer. A limited number of filters are used as the CNN performs only low pass filtering to determine the visual content of the images. After further downsampling, the 2D image is given to a neural network architecture as a fully connected layer. These five layers constitute the training process of the 2D human faces. The CNN is implemented in conjunction with the Raspberry-pi. Results of the face recognition algorithm yield a confusion matrix. Table III summarizes the performance of the proposed CNN.

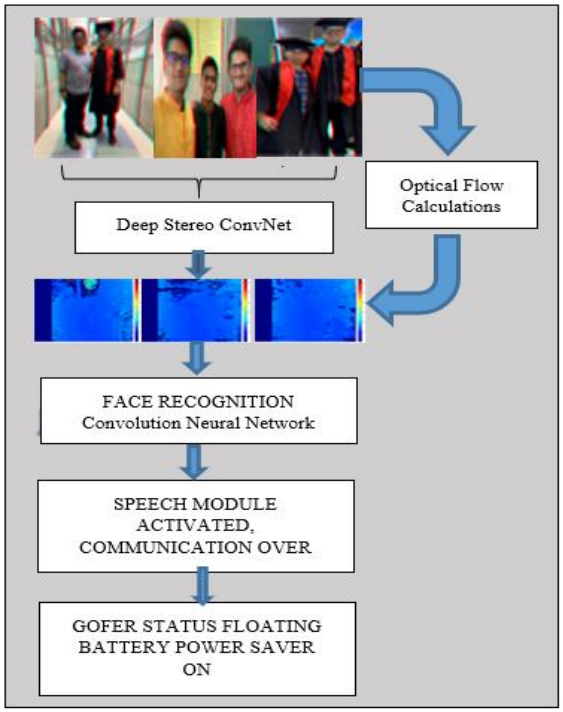

Figure 22. Journey of the Gofer, avoiding the obstacles and detecting the faces for speech communication.

The CNN provides an accuracy of 0.94. Thus, there is a 95% probability of the speech message being delivered to the correct target. Fig. 22 provides the complete flow of the process as initiated by the gofer and its execution with the appended AI.

IV. Results and Discussion

A. Simulation

The CAD model of the robot was designed on SOLIDWORKS, as shown in Fig. 23. The URDF model of this was imported in the Gazebo world, as shown in Fig. 24, on the next page. The method used was Multibody simulation where different parts (wheels, motors, plates, etc) are modelled as links, and then these links were connected using joints.

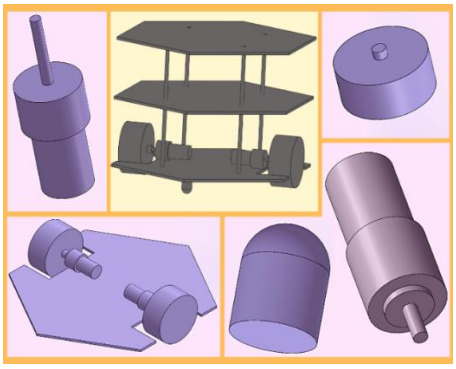

Figure 23. CAD Models of the robot assembly using SOLIDWORKS.

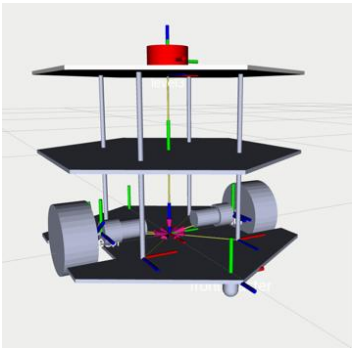

Figure 24. URDF model of the simulated bot.

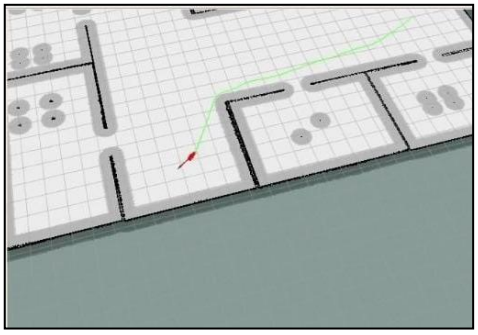

Figure 25. Path traced by the simulated robot in RViz.

A robot visualization tool named RViz shows the 2D map of the environment, as shown in Figure 25. The green line shows the path planned by the robot. In RViz, the user can customize the final destination of the robot, and the 3D Nav Goal will specify the final location.

The map was created by the robot itself using the technique called SLAM. The simulator used is Gazebo. It allows simulating our robot in various complex 3D world environments and at the same time keeping the physics realistic making it different from a video game.

There is a camera module mounted on the URDF model of the robot. In Fig. 26 we can see the video captured by it.

© 2022 Int. J. Mech. Eng. Rob. Res

Figure 26. The map of the environment.

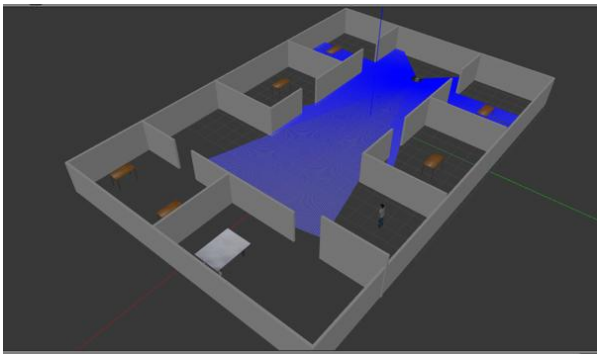

Figure 27. Gazebo World of the environment.

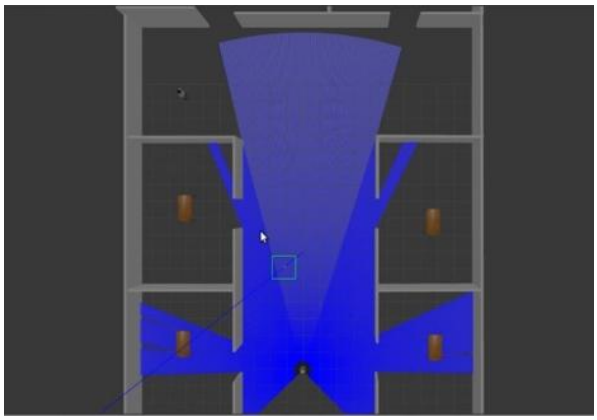

Figure 28. The initial position of the robot in simulation

This robot can be used to send messages and objects from one place to another, such as books and files. Once the robot reaches the destination, it will pass the message verbally to the person. Figure 27 shows the environment around the robot in Gazebo. Fig. 28 shows the initial position of the robot in the simulation.

To avoid the obstacles, we have proposed an obstacle avoidance algorithm. The simulated robot will avoid the obstacles, although the path planned by the AMCL algorithm is different. Fig. 29 shows the position of the robot when it has reached its destination.

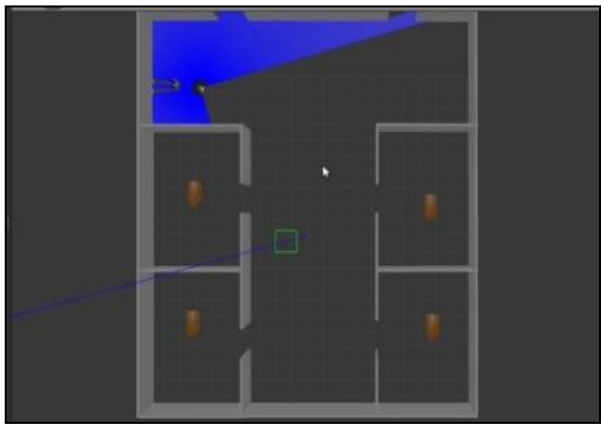

Figure 29. The position of the robot when it reaches the destination

B. Mobile Application

A mobile application is created using Flutter through which the voice commands can be sent to the robot by using Speech Recognition and manual control interface as shown in Fig. 30. It is connected to a remote database, thus making it accessible over the cloud. The user interface of the application is also shown in Fig. 30. The application will ask for confirmation of the destination. The application will send the instruction to the simulated robot if the user command is correctly recognized. The robot will follow the same steps till it reaches the destination, as explained in the simulation.

© 2022 Int. J. Mech. Eng. Rob. Res

Figure 30. Automatic Speech Recognition for Gofer User Interface with software model (above) and the Gofer Merlin (below).

The user can manually control the robot's motion using the application. The user can change the speed and direction of the moving robot. In an emergency, the user can stop the robot in the simulated environment.

Fig. 31 shows the robot in the simulated corridor environment on Blender and then imported on Gazebo. This provides more robust testing of our algorithms and a step closer to a more realistic simulation.

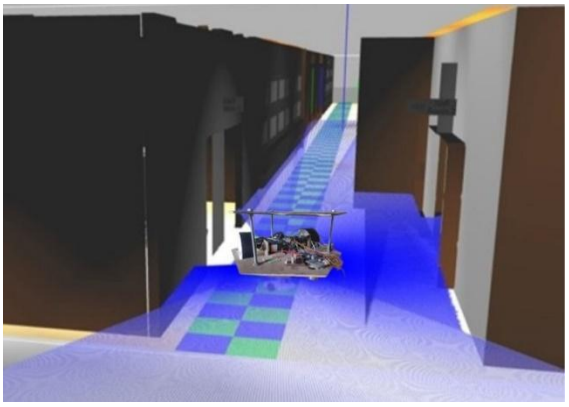

Figure 31. The bot in the simulated corridor.

V. Conclusion

In this paper, we have explained how GOFER has been built, tested, and implemented for the school subdivision on the entire floor space of over 500 sq. ft. using the fundamentals of robot kinematics and SLAM. We have explained how the sensors work so that the robot dexterously handles and maneuvers obstacles, obeys voice commands and performs functions immaculately. We have also developed a mobile application and implemented image processing to operate the robot remotely. We have designed GOFER from the chassis level using the bottom-up approach with careful load calculations and battery requirements.

This robot is a basic implementation of kinematics, computer vision, and SLAM. We can extend the robot's use in many applications, from self-driving cars to restaurants and social places to serve customers. It can also be used as an ultraviolet disinfection robot in hospitals and for sanitation purposes.

VI. Future Scope

The UGV has various applications. It can be modified to act as an Ultraviolet disinfection robot in hospitals and other public places. This UGV can serve as an industrial robot in hazardous industrial environments to reduce the human risk involved. The most common application can be for domestic use, such as an autonomous vacuum cleaning robot, or daily repetitive, mundane tasks.

We hope that this work in autonomous navigation with ROS will facilitate more advanced research in future projects and benefit the robotics community. Additionally, in implementing this project, we were able to delve deep into the inner workings of the ROS navigation, which have developed in mobile robotics such as Sebastian Turn. The practical hands-on working knowledge of ROS could prove valuable in our future careers as ROS becomes more of a standardized platform for advanced robotic applications. This project gave us working knowledge of a real-world planning system, enriching our mental toolboxes and software development skills.

Until a decade back, the bots were used for recreational activities. But now the robots are beginning to be seen in many social places. We have been hearing stories of robots serving coffee and greeting visitors at homes. These mechanical humans are here with us now. Some of the fields where autonomous robots can be used are:

- SELF-DRIVING CARS
- robotics FOR CIVIL USE
- ROBOTS IN MUSIC, SPORTS

We can extend the autonomous mapping, localization and object avoidance features of the robot in self driving cars. The accuracy given by SLAM and LIDAR is much better than GPS. This project can also be extended to charging port robots for electSric cars. If the battery of the car has low charge then the car can stop at a power station which equip such robots and the charging port robot will connect and recharge the car.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

AL and DB have worked on the hardware and chassis design of the project. AP and GK have worked on the

© 2022 Int. J. Mech. Eng. Rob. Res

software and some parts of the hardware and written the paper. SK has supervised and mentored during the research and development of the project and in the paper writing phase. All authors have approved the final version.

REFERENCES

[1] W. Y. Joe and So Young Song, "Applying human-robot interaction technology in retail industries," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, no. 6, pp. 839-844, November 2019.

[2] S. Karamchandani, S. Bhattacharjee, D. Issrani, and R. Dhar, "SLAM using neural network-based depth estimation for auto vehicle parking," in *IOT with Smart Systems. Smart Innovation, Systems and Technologies*, Senjyu, T., Mahalle, P., Perumal, T., Joshi, A. Eds., Singapore: Springer vol. 251, 2022.

[3] S. Karamchandani, S. Pednekar, A. Pusalkar, S. Bhattacharjee, and D. Issrani, "Autonomous parking system perception and control simulations on ROS-Gazebo," in *Proc. International Conference on Wireless Communication*, Vasudevan, H., Gajic, Z., Deshmukh, A.A., Eds., 2022, vol. 92.

[4] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proc. the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 340–349.

[5] M. A. Saleem, N. Senan, F. Wahid, M. Aamir, A. Samad, and M. Khan, "Comparative analysis of recent architecture of convolutional neural network," *Mathematical Problems in Engineering*, vol. 2022, pp. 1-9, 2022.

[6] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99-110, June 2006.

[7] P. Jensfelt, J. Folkesson, D. Kragic, and H. I. Christensen, "Exploiting yion," in *Proc. European Robotics Symposium. Springer Tracts in Advanced Robotics*, Christensen H.I. Eds., Springer, Berlin, Heidelberg, 2006, vol 22.

[8] M. Y. I. Idris, H. Arof, and N. M. Noor, *et al*., "A co-processor design to accelerate sequential monocular SLAM EKF process," *Measurement*, vol. 45, no. 8, pp. 2141-2152, 2012.

[9] Z. An, L. Hao, Y. Liu, and L. Dai, "Development of mobile robot SLAM based on ROS," *International Journal of Mechanical Engineering and Robotics Research*, vol. 5, no. 1, pp. 47-51, January 2016.

[10] F. Spiess, J. Friesslich, T. Kaupp, S. Kounev, and N. Strobel, "Survey and experimental comparison of RGB-D indoor robot navigation methods supported by ROS and their expansion via fusion with wheel odometry and IMU data," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 12, pp. 1532-1540, December 2020.

[11] S. Park and G. Lee, "Mapping and localization of cooperative robots by ROS and SLAM in unknown working area," in *Proc. of the SICE Annual Conference*, 2017, pp. 19-22.

[12] D. M. Turnage, "Simulation results for localization and mapping algorithms," in *Proc. Winter Simulation Conference*, 2016.

[13] H. Ibrahim M. A. Omara, K. Salleh, and M. Sahari, "Indoor mapping using kinect and ROS," in *Proc. Int. Symposium on Agents, Multi-agent Systems and Robotics*, 2015.

[14] L. Joseph. [Online]. Available: Learning Robotics using Python" published by PACKT. https://hackaday.io/project/8588-diy-autonomous-mobile-robot

[15] B. S. Sandeep and P. Supriya, "Analysis of fuzzy rules for robot path planning," in *Proc. Conference on Advances in Computing, Communications and Informatics*, 2016, pp. 21- 24.

[16] H. Malla, P. Purushothaman, S. Rajan, and V. Balasubramanian, "Object level mapping of an indoor environment using RFID," *Ubiquitous Positioning Indoor Navigation and Location Based Service*, pp. 20-21, November, 2014.

[17] Z. An, L. Hao, Y. Liu, and L. Dai, "Development of mobile robot SLAM based on ROS," *International Journal of Mechanical Engineering and Robotics Research*, vol. 5, no. 1, January 2016.

[18] K. Arai, "Moving domestic robotics control method based on creating and sharing maps with shortest path findings and obstacle avoidance" *International Journal of Advanced Research in Artificial Intelligence*, vol. 2, no. 2, 2013.

[19] S. Derek, P. Conrad, and A. Rahee, "Safe and effective navigation of autonomous robots in hazardous environments," *Autonomous Robots*, vol. 22, pp. 223-242.

[20] G. Fragapane, René de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol 294, no. 2, 2021.

[21] B. Saddek, G. Matthieu, I. François, Kahloul, Imen and T. H. Nguyen, "Designing autonomous robots. Robotics & Automation Magazine," *IEEE*, vol. 16. pp. 67–77, 2009.

[22] R. Macrorie, S. Marvin, and A. While, "Robotics and automation in the city: A research agenda," *Urban Geography*, vol. 42, 2019.

[23] H. Kerner, A. Kuntz, J. Ichnowski, and M. North, "Robotics and autonomous driving," *Technical Report Issue*, May, 2015.

[24] W. Chen, Y. Wang, H. Chen, and Y. Liu, "EIL-SLAM: Depth-enhanced edge-based infrared-LiDAR SLAM," *Journal of Field Robotics*, pp. 1-14, 2021.

[25] O. Zaki, M. Dunnigan, V. Robu, and D. Flynn, "Reliability and safety of autonomous systems based on semantic modelling for self-certification," *Robotics*, no. 3, 2021.

[26] D. V. Bui, T. Shirakawa, and H. Sato, "A UAV exploration method by detecting multiple directions with deep learning," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 10, pp. 1419-1426, October 2020.

[27] Rami A. AL-Jarrah, "Intelligent vision-based real-time detection for rough terrain navigation robot," *International Journal of Mechanical Engineering and Robotics Research*, vol. 10, no. 12, pp. 645-659, December 2021.

[28] H. Ibrahem, A. Salem, and H. S. Kang. "DTS-Depth: real-time single-image depth estimation using depth-to-space image construction," *Sensors (Basel)*, vol. 22, 2022.

[29] X. Lin, D. Sánchez-Escobedo, J. R. Casas, and M. Pardàs, "Depth estimation and semantic segmentation from a single RGB image using a hybrid convolutional neural network," *Sensors*, vol. 19, no. 8, p. 1795, 2019.

[30] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5162-5170.

[31] J. Cuevas, A. Chua, E. Sybingco, and E. A. Bakar, "Identification of river hydromorphological features using histograms of oriented gradients cascaded to the viola-jones algorithm," *International Journal of Mechanical Engineering and Robotics Research*, vol. 8, no. 2, pp. 289-292, March 2019.

Copyright © 2022 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.

Sunil Karamchandani obtained his doctorate from the Department of Electrical Engineering at IIT Bombay in Wireless Body Area Networks (WBAN).

Currently, he is associated with the Department of Electronics and Telecommunication, D. J. Sanghvi College of Engineering, University of Mumbai and is the IDC coordinator responsible for the development of innovative ideas in association with the Department of Science & Technology (DST) and National Entrepreneurship Network (NEN). His contributions are recognized in Nature India, Proceedings of IEEE, and generated extensive news coverage with leading tabloids, namely The Financial Express, Hindustan Times and Lokmat. He was a technical consultant for Blue Star InfoTech providing interactive solutions for segmentation of abdominal aortic aneurysms in collaboration with BARCO, U.K.

Dr. Karamchandani is twice the recipient of Microsoft Travel Award in 2010 and 2012. He has over fifty prominent publications in journals and conferences of repute and authored two books on Applied Mathematics over a teaching carrier spanning more than ten years. He is an avid international traveler, more recently having delivered a talk on "Promising textiles, celebrating the Silicon of the Future" at "Aurel Vlaicu", University of Arad, Romania. Senior Member IEEE.

Aayush S. Parikh earned his Bachelor of Technology in mechanical engineering with a specialization in Machine Design from Manipal Institute of Technology, Manipal, India in 2022.
Currently, he is pursuing a Master's in Mechanical Engineering from The University of Texas at Austin, United States. His most recent internship was at Mercedes-Benz Research & Development India as a student trainee, where he optimized FEA simulations to test Mechatronics systems. His previous publications include 'AI Based Nose for Trace of Churn in Assessment of Captive Customers' published in the Turkish Online Journal of Qualitative Inquiry, June 2021. His previous research includes medical robotics like 3D printed Prosthetic limbs and Ankle Foot Orthoses. His current research interests extend but are not limited to exoskeletons and human-robot interaction.
Mr. Parikh is a J.N. Tata Scholar and a part of IE Mechatronics. He has won the Best Design Award by ISIE India for his team's Solar Car.

Arman A. Lalani is a Robotics enthusiast from Mumbai, India. He has completed his engineering in Electronics & Telecommunication from Dwarkadas J. Sanghvi College of Engineering, India in the year 2021.
He was a part of the college Robotics team DJS Robocon and has participated in various Robotics competitions. He has worked on two projects which are Autonomous Bot using ROS (2020-21) and Surveillance Camera using Raspberry Pi and Simulink (2018-19)) and published respective papers in DJ Strike journals. Currently he is working as a Systems Engineer for Tata Consultancy Services, Mumbai. He works for a client in the healthcare manufacturing domain. He works in the domain of web development, cloud and system administration.

Dhruv M. Bhavsar has completed his engineering in Electronics & Telecommunication from Dwarkadas J. Sanghvi College of Engineering in Mumbai, Maharashtra, India in the year 2021.
Currently, he is working in LTI Ltd. on Java Full Stack technology as Software Engineer. He has done internships as a Software Engineer at Liminal AR VR where he has worked on Django to develop dashboard for their product ExploAR and formulated algorithms to score user-comfort and product compatibility while viewing the AR product models on their mobile devices, web development (from Trudawn Solutions in Mumbai, Maharashtra, India) where he has worked on Wix for Front-End development of a NGO website and used HTML, CSS for Front-End development of Trudawn's Website and Data Science (from Cloud Counselage Pvt. Ltd. in Mumbai, Maharashtra, India) where he has worked on Data Visualization and Data Classification using Python. He has published three papers (i.e. two on Autonomous robots using ROS (2019-21) and RFID based Smart Shopping (2018-19)) published in DJ Spark and DJ Strike journals) and has also worked on various projects in domains like Machine Learning, Data Science, Robotics and Full Stack Development .

Gaurang R. Kamat has completed his engineering in Electronics & Telecommunication from Dwarkadas J. Sanghvi College of Engineering in Mumbai, Maharashtra, India in the year 2021.
He is currently working in Infosys Ltd. on ReactJs technology as a Digital Specialist Engineer. He has done internships in web development (from Trudawn Solutions in Mumbai, Maharashtra, India) where he has worked on Wix for Front-End development of a NGO website and used Canva for making logos and Data Science (from Cloud Counselage Pvt. Ltd. in Mumbai, Maharashtra, India) where he has worked on Data Visualization and Data Classification using Python. He has worked on three projects (ie. Autonomous Bot using ROS (2020-21), Arduino Based Self Driving Car with Drowsiness Detection (2019-20) and IoT Based Smart Parking System (2018-19)) and published their papers in DJ Spark and DJ Strike journals.

© 2022 Int. J. Mech. Eng. Rob. Res