Critical Design and Control Issues of Indoor Autonomous Mobile Robots: A Review

**MD. A. K. NILOY[1], ANIKA SHAMA[1], RIPON K. CHAKRABORTTY[2], (Member, IEEE),
MICHAEL J. RYAN[2], (Senior Member, IEEE), FAISAL R. BADAL[1],
Z. TASNEEM[1], (Member, IEEE), MD H. AHAMED[1], S. I. MOYEEN[1],
SAJAL K. DAS[1], (Member, IEEE), MD F. ALI[1],
MD R. ISLAM[1], (Member, IEEE), AND DIP KUMAR SAHA[1], (Member, IEEE)**

[1]Department of Mechatronics Engineering, Rajshahi University of Engineering and Technology, Rajshahi 6204, Bangladesh
[2]Capability Systems Centre, School of Engineering and Information Technology, UNSW Canberra at the Australian Defense Force Academy, Campbell, ACT 2612, Australia

Corresponding author: Faisal R. Badal (faisalrahman1312@gmail.com)

ABSTRACT Robots that can move autonomously and can make intelligent decisions by perceiving their environments and surrounding objects are known as autonomous mobile robots. Such robots have rapidly moved from laboratories to automated industries to fill a variety of roles in our lives, homes, offices, hospitals, industries, and even on the streets. The interest in mobile robots is growing rapidly, prompting an enormous amount of research over the last 30 years, on critical factors of mobile robots such as locomotion, perception, localization, mapping, ego-motion tracking, and dynamic navigation. This article surveys these essential factors of autonomous mobile robots in terms of mathematical modeling, control issues, and challenging factors. Brief discussions are provided on the fundamentals of these technologies, popular algorithms in comprehensive mode, future challenges, and promising directions to guide the construction of an autonomous mobile robot with high accuracy and effectiveness. Since it is difficult to find complete coverage of those topics in a single location, this article provides a guideline for researchers entering the field or for innovators in the mobile robotics sector. The paper also examines open challenges in indoor mobile robots and identifies potential futures for autonomous mobile robots.

INDEX TERMS Indoor mobile robot, slam, navigation, path planning, sensors, locomotion.

I. INTRODUCTION

Robotics, which deals with the concept, design, development, control, and application of autonomous systems, is a rapidly growing sector with a wide variety of applications. Technological progress over the years has brought robotics from laboratories to almost all aspects of science, engineering, industry, manufacturing, transportation, and life in general. Even in homes, robots can perform almost any operation, including cleaning rooms [1], serving food [2], controlling doors, and serving humans efficiently [3]. Robots already make an outstanding contribution to the medical sector by serving patients [4], performing critical operations with great precision [5], controlling medical devices, and transporting and instructing patients [6]. The contributions of robotics in industrial and manufacturing sectors of industries are

remarkable in the automation of factories, and operating in hazardous environments controlling and carrying large and dangerous devices [7].

One of the rapidly growing technologies of the current world is mobile robots [9]. Mobile robots have been an interest in the research topic for decades now. It is because people imagined intelligent agents to do a lot of fascinating things & computers on wheels with simple sensors were only the beginning of these imaginations. So people researched to identify features, detect patterns and regularities, learn from experiences, localize itself, build maps of its surroundings, navigate to its desired place, and accomplish specific applications [8].

Mobile robots are able to move autonomously in the environment for which they are designed. They can sense their surrounding environments, perform actions in accordance with their instructions, and make decisions that allow them to achieve their tasks efficiently, effectively and safely [9].
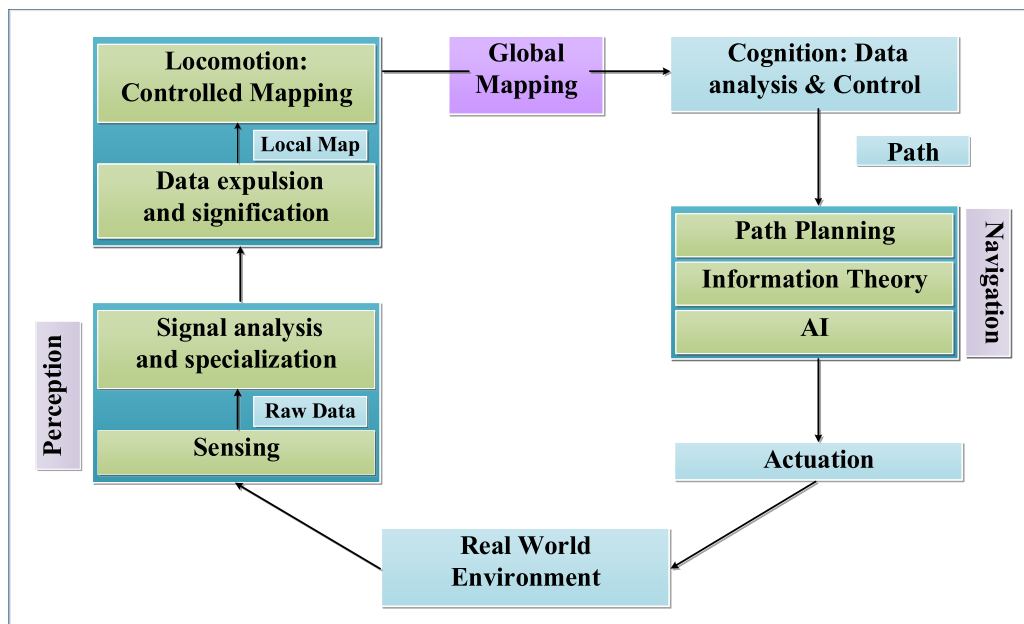
FIGURE 1. Basic Control Scheme of Mobile Robots [8].

Computing power, sensor accuracy, and powerful open-source software have led to mass production and deployment of these robots in various fields such as personal homes, automation industries, medical and health-care sectors, schools, hotel guides, construction sites, unknown land exploration, patrolling, rescuer operations and many more domains.

Any driver would prefer a less complex approach in the case of controlling the robots while handling objects [10]. The efficient design and control of a mobile robot completely depend on many factors such as locomotion, perception, cognition, and navigation. Each aspect contributes to the ability to manipulate the robots properly in the given environment [8].

Locomotion mechanisms are needed in mobile robots for them to move through their environments in order to perform their tasks that can be achieved by properly understanding the kinematics, dynamics, and mechanism of the system [11]. To understand the surrounding of a mobile robot, perception is an important factor that deals with a set of sensors whose data is extracted by the method of cognition to understand the real view of the surrounding environment [9], [12].

The selection of the best route for moving the mobile robot within the environment is done with the help of a navigation approach consisting of path-planning and object avoidance algorithms to generate real-time trajectories and avoid collisions [13]. The key factor of the navigation is localization that helps to optimize the working point of the robot within its surrounding environment [14]. The efficient control of the robot navigation largely relies on the creation of proper mapping of the environment that gives detailed information of the trajectory and obstacles within the environment [15].

Figure 1 shows the basic workflow of autonomous mobile robots using their elements. The mobile robot can be classified based on the operation to be performed. Based on its intended working environment, a mobile robot can be classified as either an outdoor or indoor mobile robot.

Outdoor mobile robots are mainly used in land exploration with the help of global positioning system (GPS) sensors to localize and navigate autonomously. They are used in the agricultural field to automate traditionally labor-intensive agriculture practices, in defense, to assist the military forces to find bombs, harmful objects, or navigate and to gather intelligence [16]. The controlling of locomotion is a critical challenging task for outdoor mobile robots due to the dynamics of landscapes. Indoor mobile robots have developed rapidly from simple line followers that are restricted to move within a fixed-line, to robots that can recognize not only their surroundings but also adjust to accommodate dynamic objects such as people around them. The main challenges for indoor robots are localization and positioning as GPS systems do not work indoors [17]. Simultaneous localization and mapping (SLAM) require high-level sensor fusion techniques with visual perception capabilities. With the exponential advancements in computer science, particularly the growth in artificial intelligence, along with the increase in computing power in edge devices, the efficiency of indoor robot navigation has increased dramatically after 2010 [18], [19]. Some uses of robots in indoor spaces include personal assistants [3], industrial robots [20], waiter robots [21], a personal assistant robot [22], take-care robots [23] etc. are becoming true. Figure 2 is evidence of the increasing popularity of indoor mobile robot throughout the world.
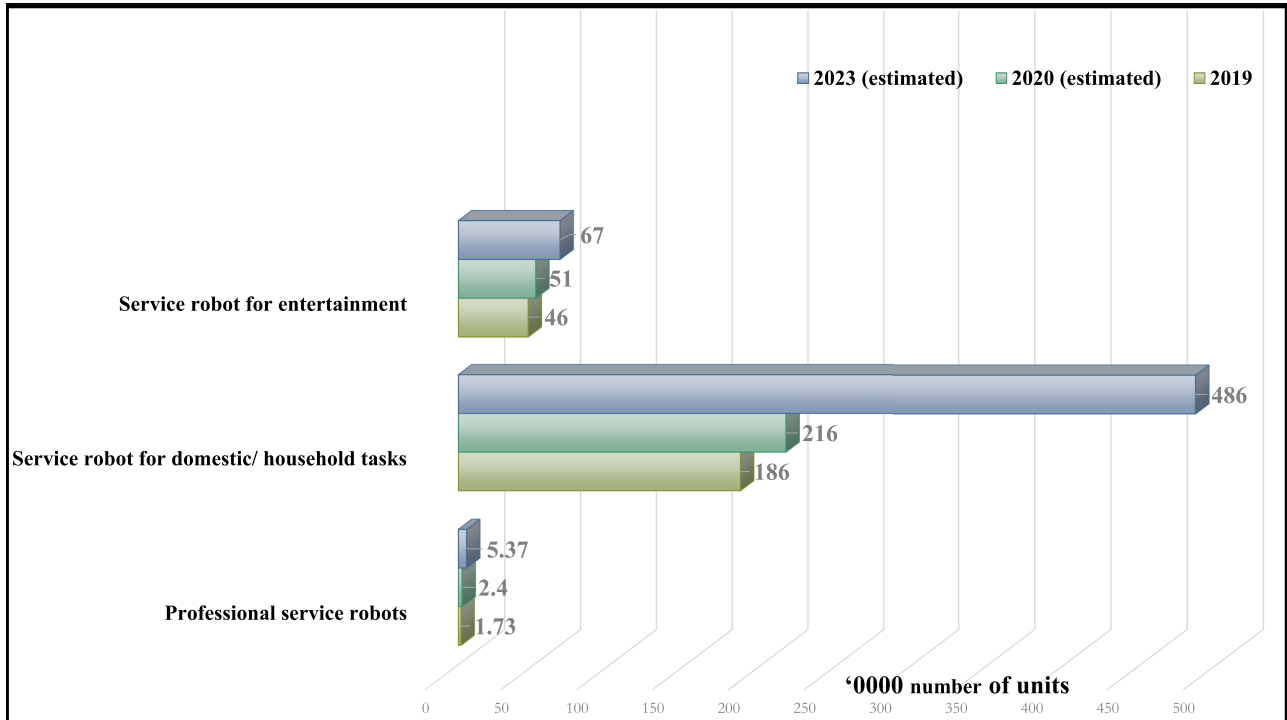
FIGURE 2. Projection of Number of Robots Manufactured for Different Applications [16].

Mobile robotics is based on a broad range of algorithms and technologies which have progressed dramatically in the last twenty years. Cadena *et al.* [24] have provided a survey of SLAM, explaining open challenges in SLAM and mapping. However, the article did not cover the recent advances of SLAM using deep learning, which has been covered by [19]. This article has surveyed the deep-learning based solutions for localization and mapping, while fundamentals like locomotion and navigation have not been covered. An excellent survey on navigation techniques such as obstacle avoidance and path planning is provided by [18]. Alatise *et al.* [12] discussed the methods, challenges, and scope of sensor-fusion methods in mobile robotics broadly, but did not discuss other aspects of mobile robotics. These papers are focused on specific aspects of mobile robots and do not provide all of the fundamentals that make the mobile robot reliable and controllable.

The efficient regulation of the indoor mobile robot largely depends on the proper control of locomotion, perception, cognition, navigation as well as mapping. The aforementioned research articles have focused on only one control side of the mobile robot. Detailed information of all the controlling factors of the mobile robot is missing in these research articles. The main motivation of the research article is to provide more detailed information about all controlling factors which makes the design and control procedure easy and efficient. The fundamentals of the mobile robots such as locomotion, perception, navigation, localization, mapping, path planning, and obstacle avoidance as shown in Figure 3 are the essentials

FIGURE 3. Paper Organization.

aspects of mobile robots which are reviewed in this article in order to provide a comprehensive guide to the researcher. The modeling and control issues of these fundamentals are investigated in terms of mathematical modeling, efficiency, control approach, and computational cost estimation. The available technologies of these fundamentals are analyzed in terms of their integration and design complexity. The minimization of

the complexity is reviewed by describing control technologies available in the research community and industry. Different advanced control technique such as machine learning and deep learning is becoming popular due to their high level of accuracy and autonomous control technique. The article also reviews the implementation of deep learning approach in different controlling factors of the mobile robots to explore the new research area in robotics.

A. PAPER ORGANIZATION

The paper begins in section I with a standard formulation of indoor autonomous mobile robots and a brief introduction to their core elements. Section II then begins to address core concepts starting with locomotion in mobile robots, discussing the type of locomotion, use cases, advantages, and limitations. Section III formulates a general concept of how mobile robots perceive the world using sensors, as well as the type of sensor and sensor-fusion methods that can be used. Section IV discusses cognitive models of mobile robots and section V introduces the general components of navigation, with subsection A discussing the kinds of maps generated for mobile robotics applications, and subsection B addressing the basic paradigms of localization providing a comparison. Subsection C overviews SLAM algorithms, SLAM's importance, SLAM paradigms, and recent advances in visual SLAM. Subsection D and E outline cognition schemes for a mobile robot through its path planning and obstacle-avoidance strategies. Section VI highlights recent advances in mobile robotics with the power of deep learning. Section VII discusses open challenges and future trends in mobile robotics and final remarks are in section VIII. Figure 3 provides an illustration of the organization of the paper.

II. LOCOMOTION

Mobile robots need to be able to work in any environment including hazardous, rough, loose, uneven terrain, surrounded by obstacles and taboo areas. Locomotion is therefore fundamental for a robot's movement through the working environment, and not only depends on the nature of the environment or path of the robot's movement but also on vital factors such as the required ability to be controlled, the degree of maneuverability, efficiency, and stability in varying terrain situations [9]. Indoor mobile robots are designed to move, run or walk in order to complete their specified assignments, so can be equipped with legs or wheels or wings sometimes (wheeled robots are proven to be most efficient) [25]. These various multi-terrain robot locomotion procedures have been discussed in this article.

A. GAIT LOCOMOTION

GAIT locomotion implements one of the locomotive mechanisms from nature, such as biped, quadruped, and hexapod. According to one unique method, the leg configuration can be devised on a pantograph mechanism which is spring loaded with multiple segments [36].

The legged locomotion allows the robots to be high in mobility, keeping both dynamically, and statically stable gaits and overcome significant obstacles such as several steps or stairs [37].

The complexity of the limb coordination of any robot depends on the number of legs, which defines the GAIT pattern [25]. If the robot is built with n number of legs then it has the $(2n - 1)!$ possible number of events [8]. The leg configuration can be described in various ways. Table 1 discusses the applications and limitations of different gait locomotion methods and each is described in more detail in the following sections.

1) ONE-LEGGED ROBOT (HOPPER)

A robot can have at least one leg, which is also called "hopper" because there is only a single contact point between the ground and foot and the robot has only one kind of motion which is hopping. This comes with the manner of duo subdynamics; one when contact between the leg and ground are in contact and another for contactless situations. [38]. One vital application of these hopper robots is in the research of small celestial bodies such as asteroids and comets where multi-legged or robots with wheeled locomotion are not able to maneuver successfully as the local gravity is relatively low [39].

2) TWO-LEGGED ROBOT (BIPED ROBOT)

Two-legged design is most commonly used in robot locomotion. These two-legged robots, also known as humanoid robots, are able to walk, jump, run and sometimes dance. They can climb stairs and traverse different rough surfaces but dynamic stability is still an issue. Biped robot locomotion has shown several advantages over single or multi-legged locomotion in many research cases [40], due to the dynamic balance.

3) THREE-LEGGED ROBOT (TRIPOD ROBOT)

Three-legged robots are rare as they aren't inspired by any biological examples. The only way these robots can make movements is by moving their center of gravity out of the body keeping the action balanced [25]. Three-legged robots have better mass balance than two-legged in more rugged surface areas.

4) FOUR-LEGGED ROBOT (QUADRUPED ROBOTS)

Quadruped robots are the most easily balanced, having the best dynamic stability. These animal-like four-legged robots are also popular because they can be treated as pet substitutes. There are also six-legged robots called hexapods. In [41], various control architectures are applied in the four-legged robot to monitor robot behavior. These procedures are base on bio-inspired structures help in carrying heavy loads for their dynamic stability and heavy equipment carrying capacity through the rough surface [41].

TABLE 1. Summary on Different Types of Gait Locomotion.

Locomotion Type	Ref	Usefulness	Applications	Limitations
One legged robot (Hopper)	[26]	(i) Can jump over obstacles more easily (ii) Are able to position themselves less concern about static stability (iii) Requires only single contact point on the ground to maneuver over rough terrain (iv) No requirements of leg coordination.	This locomotion system can be used when the robot is move in places where jumping over obstacles is needed, for example in the stairs, rough geographical topographies and small celestial bodies in outer space.	(i) Needs to be balanced especially in abrupt stationary situations. (ii) Static stability is near to impossible.
Two legged robot (Biped)	[27], [28]	(i) Dynamic balance is better and easier than hoppers to establish stability (ii) Are able to move over irregular surface efficiently (iii) Leg coordination is easier than multi-legged locomotion.	This kind of locomotion system is mostly applied in humanoid robots assisting humans.	(i) The size and weight of the actuators are two significant botheration while constructing a biped robot [29] (ii) There is still an issue with the dynamic balance but in recent research this issue can largely be overcome.
Three legged robot (Tripod)	[30]	(i) A tripod offers advantages in balancing while walking and even when stationary (ii) Can easily change their direction in the trajectory (iii) Can move in both circles and line path according to the need.	These robots can be used in military applications to carry heavy weight as they have good dynamic balance.	These three-legged locomotion is quite difficult to find in nature so some biological complexity is still present.
Four legged robot (Quadruped)	[31], [32], [33]	(i) Dynamically most stable robots [34] (ii) Can balance by estimating their acceleration along with lateral velocity with the help of the sensors in the legs and this helps them to balance dynamically (iii) Can carry more weight than other designs (iv) Can maneuver through rough terrain while walking, running, climbing and also carrying heavy loads [35]	These are largely used in military applications as they can carry load in rough terrains easily and also can maneuver in environments that are hazardous to humans. They can also be treated as pets & emotional relationships can be developed which can be an alternative to a real pet.	(i) The leg combination while walking is more complex than the other locomotion systems.

B. WHEELED LOCOMOTION

Wheeled locomotion is one of the most popular mechanisms as it is the easiest means of locomotion for mobile robotics as it is for many man-made vehicles generally. Wheeled locomotion has achieved very good efficiencies due to its stability, easy controllability, and also relatively simpler mechanical design. Balancing is not really a major concern as the design of wheels guarantee that it is always in contact with the ground thus providing the maximum stability and low acceleration rate, which helps them avoid moving too far too quickly [42]. There are four different types of wheeled locomotion:

1) The Standard Wheel: two degrees of freedom and rotation occurs around the wheel axle and point of contact.
2) The Castor wheel: two degrees of freedom and the turning occurs around the offset steering joint.
3) The Swedish (or Mecanum) wheel: three degrees of freedom and the rotation occurs around the rollers, wheel axle and contact point.
4) The Ball or Spherical wheel: has more surface area than the previous cylindrical castor wheels.

Figure 4 portrays all categories of wheels applied in mobile robots. Depending on the environment, the wheel arrangement is considered simultaneously while choosing wheel type [43]. Both the wheel type and wheel geometry play vital
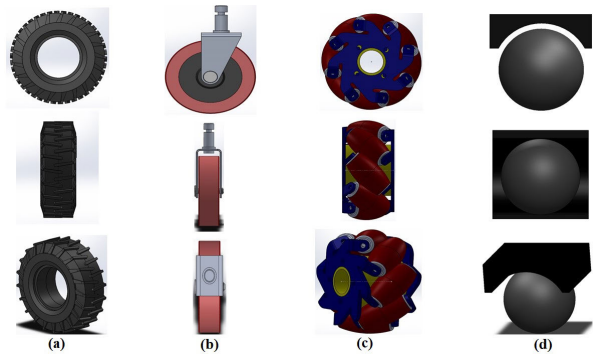

FIGURE 4. Four Types of Wheels (a) Standard Wheel, (b) Castor Wheel, (c) Swedish Wheel and (d) Ball wheel [8].

roles in the fundamental robot characteristics; maneuverability, controllability, and stability. Stability requires two wheels minimum if the center of gravity is directly below the axle of the wheel, but a two-wheeled robot can strike the ground, for instance, when in motion. Generally, three-wheeled robots are the most stable statically with the center of gravity in the center of the wheels. For further improved stability, the wheel number can be increased. Maneuverability is the ability to maneuver in any direction gradually [25] and is one of the vital concerns of the robot design. Table 2 compares the usage and limitations of the different kinds of wheels in mobile robots.

TABLE 2. Summary on Different Types of Wheel Locomotion.

Wheel Type	Ref	Application	Usefulness	Limitations.
The Standard wheel	[44]	Standard wheels are applicable in almost every type of vehicle: cars, motor cycles and also in mobile robots to maneuver rough terrain.	(i) Can accomplish steering motion in different direction without any kind of side effects. The steering action of this wheel can resist self movement of the robot chassis	Limited motion because of the condition of no side slip.
Castor wheel	[45]	(i) This type of wheel is used in a wide range of applications from office chairs to mobile robots. (ii) This wheels are also used in medical field, industrial field and educational field.	Castor wheels are one of the most significant solutions to providing omnidirectionality.	i. As this type of wheel tends to rotate around the offset axis, it causes an imparted force on the robot chassis while steering. ii. This wheel produces comparatively complex kinematics which makes it difficult to program.
The Swedish or Mecanum wheel	[46], [47]	(i) Better maneuver in rough terrain in omni-direction. (ii) Used in both outdoor and indoor applications including searching and rescuing missions, many military activities, mine operations and also in planetary explorations, shopping carts or powered robotic wheelchairs.	(i) It is basically used in efficient omidirectional movement. As the robots with this kind of wheel are able to have full mobility over the plane which means the wheel helps them to move in each direction at any instant without facing any reorientation. (ii) The design of wheel is compact and has higher load capability. (iii) Controlling this wheel drive is easier.	i. Design is conceptually very complex. ii. Robots face some discontinuous contact of wheel and higher sensitivity towards floor irregularities, hence more torque is needed from the motor than standard wheels.
The Ball or Spherical wheel	[48], [49]	(i) These wheels are applicable in wheels of supercars, office chairs, mobile robots, mars rover locomotion. (ii) Robotics ballbots are specific kind of application of spherical wheel.	(i) Can be moved at any direction directly (ii) Is able to include motion from point to point and also keeping station along with dynamic balance. (iii) Can also have movement in arbitrary direction.	There are some design issues along with technical problems regarding this spherical wheel such as tucking this wheel tire beneath any car's fender requires more space, which means that the front and the rear axles must be shortened considerably.

C. OTHER UNCONVENTIONAL CONCEPTS

Along with the most commonly used types of legged and wheeled locomotion, there are some other types used. The concept of Serpentine Locomotion is taken from the movement of a snake to push the robot's body forward [50]. Track Belt/Slip Locomotion is another technique that overcomes the problems faced by wheeled locomotion to move in soft terrain and has dynamic stability due to its large surface area [51]. Wheeled and legged locomotion are also combined to overcome the drawbacks of both types in what is called the Legged Wheeled locomotion or Walking wheels [52].

These locomotion techniques help the mobile robots maneuver through their indoor environment. Although no one mechanism is suited to all environments, wheeled and legged locomotion systems have many successful applications [25].

III. PERCEPTION

Perception relates to the ability to see, hear, or become aware of the surrounding environment. In robotics, especially for a self-governing mobile robot, perception means the ability to attain knowledge about itself and the external context within which it is located [53]. For a robot to work autonomously, perception is vital and is achieved with the help of high-resolution sensors and algorithms to extract information from them [54].

Figure 5 shows the basic flow diagram of the perception process.

The main components of perception are:
- Processing data from sensors
- Representation of data (environmental modeling)
- Artificial Intelligence Algorithms

Perception helps the robot to operate numerous functions [55]:
- Detection of human, vehicle and obstacle detection [56]
- 2D and 3D environment
- Recognition of gesture, voice and some other activity and semantic place classifications
- Detection of environmental changes
- Classification of terrain [57],

All the examples are achieved with the help of some specific and highly precise sensors such as various vision sensors and proximity sensors [9].

A. SENSORS

A sensor can detect physical changes in the environment and transform the changes into electrical signals that can be used in further procedures. Sensors can be categorized into two broad categories [12]:

1) Proprioceptive or Exteroceptive kinds of sensors
2) Active or Passive sensors

A proprioceptive sensor is able to read the internal changes of the robot such as the internal battery voltage and temperature, speed of motors used, acceleration, and net loads
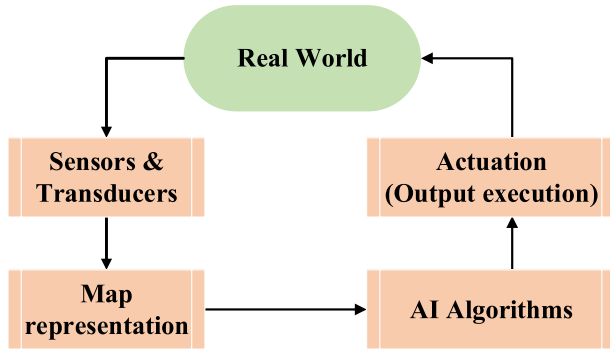
FIGURE 5. The Main Components of Perception [58].

on the wheels. Exteroceptive sensors measure aspects of the external environment of the robot. An active sensor has its own source of energy, such as an IR proximity sensor, which is emitted and the return measured in some way. A passive sensor can detect energy in response to some kind of inputs from the internal or external environment such as gyroscope motion [11]. A robot will be designed with selected sensors based on its role as an indoor or outdoor robot as well as its allocated tasks.

1) RGB-D SENSOR

RGB sensors are color sensors that are used to detect the color of any surface in the RGB (red, green, blue) scale. An RGB-D camera that can simultaneously report the information about color and depth is a very good solution for needing accurate objection detection. The camera uses in-depth technology measurements and can use a single sensor to measure RGB and depth with high robustness with an excellent signal to noise ratio [59]. One of the cheapest options for an RGB-D sensor is the Microsoft Kinect shown in Figure 6.

2) INERTIA MEASUREMENT UNITS

The basic inertial sensors are combined as Inertia measurement Units (IMU) comprising various sensors such as accelerometers or acceleration sensors, gyro sensors or gyroscope, and magnetometers [12].

a: ACCELERATION SENSOR

An accelerometer can be used to estimate the angle of any vehicle but has great difficulty in detecting the difference between the vehicle's acceleration and the acceleration of gravity. For the axis along g_x and g_y, the acceleration angle [44] is:

$$\theta = (tan)^{-1} \frac{g_x}{g_y} \tag{1}$$

Which illustrates that angle estimation accuracy decreases if the vehicle increases or decreases its speed.

b: GYRO SENSOR

A gyro sensor or gyroscope is a mounted wheel with two or three gimbals and pivoting supports which allow wheel
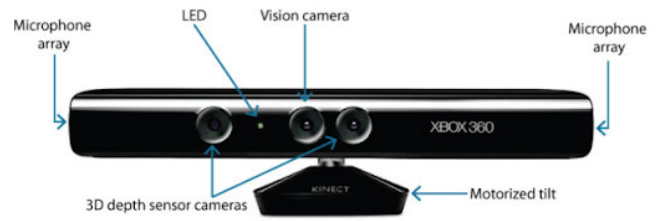

FIGURE 6. Microsoft Xbox 360 Kinect [64].

rotation about a single axis. One gimbal is seated upon the other with the pivot of orthogonal axes. The other kind of gyroscope has two gimbals one of them is the outer gimbal and the other is the inner gimbal. For gyro angle [44]:

$$\theta = - \int \omega dt \tag{2}$$

where ω is the angular velocity.

c: MAGNETOMETER

Magnetometers are able to measure the strength, direction, and also change in the magnetic field (like a compass). There are combinations of magnetometer and accelerometer, called an accelerometer-magnetometer combination (AMC), that can be used to extract angular rate, sensing motion or in gesture detection, and so on in the concept of virtual gyro in robotics [60].

3) MOTION SENSOR

In object detection techniques, motion sensors play a vital role [61]. These devices can be often integrated with some other comparative components and used as a system that automatically can perform a task such as alerting the user about a motion in a certain area.

4) BIO-POTENTIAL SENSORS

Bio-potential sensors are built with transmitters with confined range and power, used for transmitting biopotential information in form of signals. An algorithm then is operated to compute the signals with the help of the previously recorded value and the measured value. These sensors can be used in an integrated form in making some kinds of wireless headsets [62] and measuring signals from any patient's skin.

5) ENCODERS

An encoder is a kind of electro-mechanical device that can provide motion control with position, velocity, and change in direction. Encoders are of two types; the linear encoder which response to motion along a linear path, and the rotary encoder which response to rotation. Encoders can be connected to the controller for data analysis [63].

6) INFRARED SENSOR

An IR sensor [65] measures the heat energy emitted from an object body and can detect motion of that object [63].

7) OPTICAL FLOW SENSOR

Optical flow sensors can be used to make texture and features visible to determine ground velocity. These sensors can be used in projects where weight and size are issues.

8) LIDAR SENSOR

LIDAR, the acronym for ''Light Detection And Ranging'' is one kind of sensing method, which is often called a laser scanner or 2D LIDAR sensor as shown in Figure 7. LIDAR sensor is one of the most used sensors that measures distance with the help of illuminating a target by laser light and also measure the reflected light with a sensing element [66].

FIGURE 7. Rplidar A1 by SLAMTEC [71].

9) PRESSURE AND FORCE SENSOR

The measured force or pressure can be converted into a change in length or other dimensions of a material or spring element. This change can be measured by a sensory element such as a piezo-resistive gauge or any resonant strain gauge or varying capacitance [67]

10) ULTRASONIC SENSOR

Ultrasonic sensors are able to measure distance with the help of ultrasonic sound waves. The sensor can quantify the time between emitting the wave and receiving it back, from which it can calculate the distance [68].

B. FUSION TECHNIQUES OF SENSORS

Practically, when sensor data is analyzed, there are many sources of noise and uncertainty [69] such that multi-sensor integration is almost always useful to fuse or combine various sensory information sources into a single representational format. A multi-sensory data fusion (MSDF) system can combine all the complementary sensors, additional sensors, or even from only one single sensor within the constraint of a programmable amount of time [70]. This kind of combination of sensors provides a number of advantages as summarized in Table 3.

Searching methods fall into two basic categories [8].

1) BASED ON AREAS

These types of algorithms mostly identify small portions of the whole image and seek a similar area in the sample image, by measuring the appropriate correlation [8]. There are several methods that are being used for the measurement of the

uniformity of the image areas for stereo matching such as the Sum of Absolute Difference (SAD), the Sum of Squared Difference (SSD), and Normalized Cross Correlation (NCC).

2) BASED ON FEATURES

Feature-based
algorithms extract prominent features present in the images. These sensors usually consider the corners, edges, blocs, and line segments as the matching features, which are not necessarily to be a geometrically well-defined entities. There's a drawback that these only provide sparse depth maps which must be interpolated. The fusion methods can also be categorized as follows [72]:

a: STATE ESTIMATION METHOD

These types of methods are usable as the data can come from any sensor or be at any state which is unpredictable. The Particle Filter and Kalman Filter are two basic state estimation methods. The Kalman filter estimates some unknown state of the systems of various fields [12] for object tracking, navigation and localization, and the particle filter build a representation based on Probability Density Functions (PDFs).

b: DECISION FUSION METHOD

This method integrates the decisions from multiple classifying units into a single efficient decision about the activity that occurred in the area. This method is able to reduce the uncertainty level by measuring the evidence at a maximum level. One of the two basic types of decision fusion approaches is the Bayesian Approach, which uses the Bayes Theorem to update the probability for any system where more information or evidence is available for analysis. The second type is the Dempster-Shafer (DS) which is widely used in signal recognition and solving with better adaptability in unknown and uncertain territory problems [12].

IV. COGNITION

Whenever the robot is embedded in an environment, it accumulates data about its surroundings and collects data about the route, and becomes more and more familiar with everything around. It can identify the places it visited before.

A mobile robot comprises two vital sections; the mechanical structure and the control section. The control section, grounded on perception, processing, and cognition-navigation, is used to monitor the mechanical construction's ability to achieve the required objectives. [9] Perception offers the ability to observe, hear and generate awareness of the surroundings. [53] The information gathered by perception is then refined and adequate injunctions are received by actuators for further movement. After processing, the cognition function identifies the approach and objective. The main focus of cognition is planning an effortless reference track from the initial count to the destined mark avoiding collisions in the working environment [75]. To constitute the representation of the robot, its surroundings, and interaction system, a 'cognition model' is needed and tracking using

TABLE 3. A Comparative Table of Different Sensor Fusion Techniques.

Sensor Type	Sensor name	ref and year	Fusion Method	Efficiency	Limitations	Advantages
Inertia sensors	Acceleration sensor & gyro sensor	[12], 2020	Low pass filters	The inertia sensors are of medium accuracy. For cases such as vehicles accelerating at very high or low speeds, the accuracy decreases.	(i) These are not able to differentiate between gravitational and vehicle's own acceleration. (ii) Filters are used to reduce the noise.	These sensors are low cost with two-axis acceleration and one-axis gyro sensor.
Inertial measurement unit (IMU)	Acceleration and Gyro sensor	[12], 2020	Kalman Filter (Complementary)	Shows accuracy after calibration with the complementary filter.	When the vehicle speed is very high, the embedded filters are unable to reduce the full noise. A complementary filter is needed and this increases the cost.	(i) These sensor units are very low in cost and power and high in performance. (ii) Easily usable in smartphones and smart devices such as mobile robots.
Inertial Measurement Unit	Acceleration and gyro sensors, Magnetic Encoder (Non-contact type)	[9], 2014	Sophisticated Estimation Filter	These sensors provide high efficiency and accurate output.	The cost is pretty high for low budget robots.	These sensors provide a wide range of measurement. Also attitude and navigation problems can be solved.
Inertia & Tactile Sensors	Accelerometer, Position and Force/ Torque sensors.	[73], 2002	Kalman Filter	Provides medium efficiency according to the calibration, but with the help of better filter algorithms, the noise can be filtered, and the efficiency can be increased.	These sensors alone can not provide proper posture control. Some algorithms such as Kalman estimators are needed to be integrated to get better results.	These sensors are essential for manipulation tasks and also for sensing the state of the part to control the posture.
Encoder & IMU	Three axis Gyroscope & Accelerometer IMU and Wheel encoder Sensor	[74], 2019	Extended Kalman Filter	Significant improvements in odometry accuracy have been seen involving the fusion job. The error in achieving the final position has been reduced. Overall localization with fusion has been improved from single sensor usage.	(i) The localization area is limited to within a few centimeters. (ii) The sensor frequency and resolution is still a challenge to improve.	The mapping error percentage is reduced significantly (0.021%). Great improvement in odometry exactitude is shown.

pattern recognition, including computer vision. For tracking purposes, control is needed, and a performance error can be identified [9] which requires various control techniques.

V. NAVIGATION

Navigation provides the toughest challenges for mobile robots, particularly for indoor navigation. For a successful path navigation approach within the territory, a mobile robot must simultaneously perform localization and keep track of its surroundings–simultaneous localization and mapping (SLAM) [9].

Navigation can be broadly classified into four major concerns: i. Environmental Modelling (Mapping), ii. Localization, iii. Path planning, iv. Obstacle avoidance [12].

A. MAPPING

The fundamental problem of mobile robotics often concerns two questions, where is the robot at a given time, and what is its surrounding environment. For the robot to navigate smoothly while achieving some desired task, there are several methods but the best approach is the map based

approach. Mapping basically means creating some representation that the robot uses to model the environment, gain knowledge about the environment, and take decisions based on that knowledge, and achieve its goal [12]. The problem of finding the position and modelling the surroundings around the robot is a dual problem that can be related to the chicken-egg problem. So often the problem is defined using **SLAM** [76]. SLAM means computing the different robot positions as well as the model of the environment simultaneously. Three vital relationships are to be considered while choosing a certain map representation [8]:

1) The map's precision needs to match the desired precision of the robot while achieving its goal.
2) The features represented by the map must match with the features that are extracted by the sensors.
3) The arithmetic efficiency of mapping, localization, and navigation is directly affected by the complexity of the map representation.

In the world of mobile robotics, two types of traditional map representation have been popular: metric map and topological map [77]. Metric & topological maps are

generally associated with 2D map representations. Different types of 3D map representations have surfaced in the last decade including a sparse map, semi-dense map, dense map [78]. Semantic maps can be generated in both 2D [79] & 3D [80] representation.

1) METRIC MAP REPRESENTATION

A metric map shows the geometric attributes of a surrounding environment. Generally, in a metric map, a location or an object is presented as a co-ordinate. Idiothetic data plays a vital role in this approach and the raw sensor data is converted into information regarding the 2D space. Metric models usually are highly accurate in presenting the objects in the surrounding environment and easy to understand from a human point of view [77].

Metric maps depend heavily on Idiothetic data, but in reality, are prone to cumulative error. The quality of such data decreases as time goes by, so this data cannot be trusted after a certain length of time. Also, the model relies on the accuracy of sensors and the circumstances around them.

Two popular metric map approaches are Feature-based map representation and Free-space representation [81].

a: FEATURE-BASED MAP REPRESENTATION

A feature map representation annotates the positions of the features of the environment with precision in a continuous space. Most indoor mobile robots use sonar or lidar sensors to learn the distance between themselves and the accessible objects. These robots can bring out the best lines from the integrated sensor data. Using such lines, objects or boundaries might be represented in a metric map [82], [83]. This process represents the physical location of the walls or any other objects in the environment without adding any additional features such as color, texture, etc.

b: FREE-SPACE REPRESENTATION

This approach deals with the free space that the robot is able to access. This simplifies the approach of adding different features, rather it just maps the free area in the surrounding access to the robot. One of the most common and used map representation technique in mobile robotics is occupancy grid representation, which follows the free space approach [84].

In the occupancy grid, the circumstance is divided into a distinct grid consisting of two kinds of cells. The cells that contain objects are called occupied cells, and the other cells are a part of free space. The cell's probabilistic value increases with increasing hits, and after a certain threshold, it is determined as an occupied cell [85]. The significant advantage of this approach is it can straightly use sensor data without the necessity to extract any feature [81]. Figure 8 shows a basic representation of the feature-based map and occupancy-grid map for a real-world environment.

2) TOPOLOGICAL MAP REPRESENTATION

The topological approach doesn't concentrate on exact geometric co-ordinates of the environment; instead, it focuses
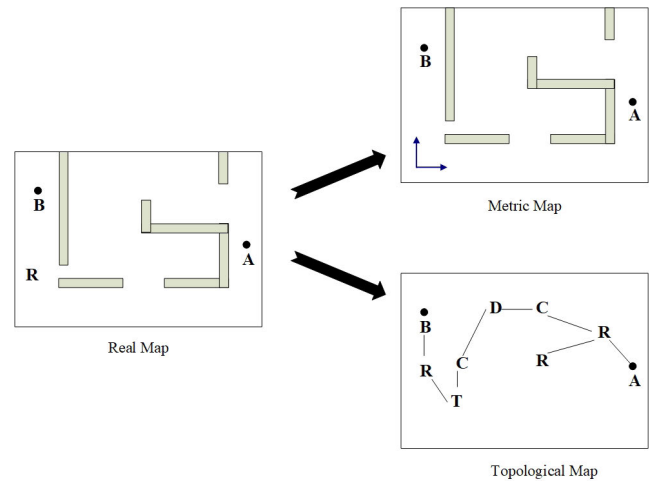

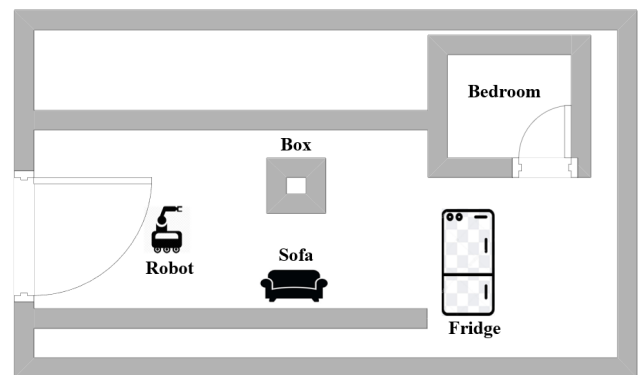**FIGURE 8. Metric vs Topological Map Representation [81].**

FIGURE 9. Semantic Map.

on distinct characteristics of the most relevant environment for the robot to localize and navigate effectively [77]. The topological approach consists of nodes and arcs. Nodes are referred to as locations in the environment that the robot can traverse and arcs are used to connect adjacent nodes.

The main benefit of topological representation is that it doesn't require the transformation of the sensor data in a 2D reference frame [81] but only requires the storage of location information given a sensor reading. The drawback of this representation is that the sensor information that defines key places is only available if the robot has physically visited that place before, meaning that more expensive exploration is needed of the environment in order to estimate position at higher precision. Figure 10 shows a basic representation of the metric and topological maps.

3) SPARSE MAP REPRESENTATION

Sparse maps are generated through feature-based SLAM methods [78]. These methods extract visual features from the frames captured from the camera and construct the map using the set of key features needed to describe the scene geometry. A sparse map consists of persistent keyframes in a geometric scene. These methods check for common map points in
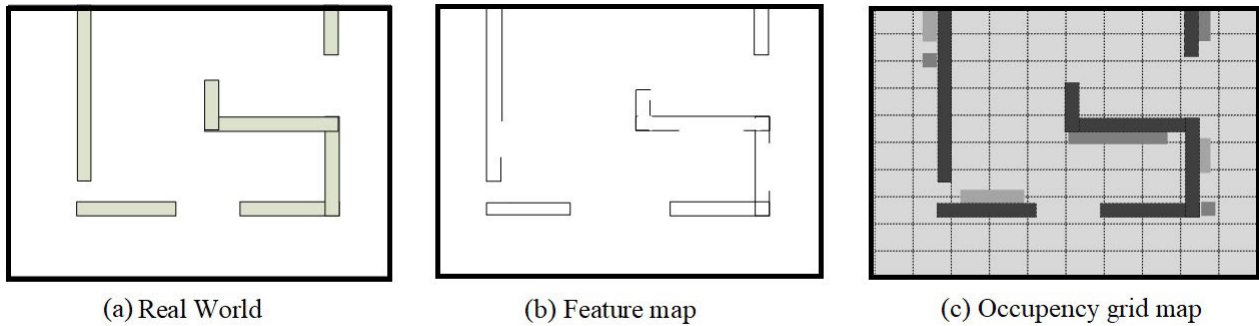
(a) Real World   (b) Feature map   (c) Occupency grid map

FIGURE 10. (a) Real World, (b) Feature-based Map and (c) Occupancy-grid Map [81].

different timeframes and fix a set of active keyframes in a fixed location containing rich geometric information about the location [86]. As sparse maps store only the active map points required to form a 3D representation, it is generally less computationally expensive than generating semi-dense or dense map representation.

4) DENSE & SEMI-DENSE MAP REPRESENTATION

While sparse maps are very useful for reducing computational complexity, and visual localization; these maps are often not very helpful for robotics applications such as navigation or interaction with surrounding objects [87]. Dense maps provide far richer geometric information from images, helping robots interact with their nearby objects & take intelligent decisions. Dense maps are created by capturing all of the geometric information available using visual sensors, and reconstructing the structural nature of the objects [78]. Though, dense maps are computationally very expensive; hence, they require the use of a powerful GPU for real-time inference. Semi-dense maps reduce this complexity to multi-thread operation on CPU only by building a depth map using more key-points than sparse map [87]. This results in a map that can hold more geometric information and helps robots interact with objects far better, which, eventually helps in succeeding in various robotics tasks such as navigation.

5) SEMANTIC MAP REPRESENTATION

Semantic maps use contextual clues to develop a qualitative description of a robot's surroundings and use that information to understand complex unstructured environments. To achieve this, the robot must retain a cognitive interpretation of space. This process should involve semantic attributes about objects and places encountered correlated to the surrounding's geometrical perception. Figure 9 shows a semantic map generated by [79].

Since Metric and topological maps only describe spatial information regarding the surrounding environment, additional information might be needed for the execution of multiple and complex robotic tasks. Semantic maps enlarge the scope of adding additional information to maps such as instances, categories, and attributes of an object as well as

common sense knowledge regarding the entities present in the maps.

Semantic maps may vary in nature but they should always include [88]:

- Spatial information contained in the map such as metric or topological map
- Anchoring of spatial concepts such as place or object classification

The drawback of a semantic map is that, as it holds a high level of information, it has high memory usage [89]. To do a cognitive interpretation of the environment, a high level of object detection is necessary, which wasn't available, at least until the late 2000s.

B. LOCALIZATION

Localization has often been the most challenging problem in mobile robotics. It is essential to figure out where the robot is and whether the absolute map of the environment is available or not. Three types of localization problem have been described by [90]:

- Local pose tracking: The robot's previous position is needed to track the current location which is solved by modelling the robot belief with a unimodal distribution such as the normal distribution [91].
- Global Localization: This assumes the robot localizes concerning the map with the help of a uniform distribution as the initial belief of the robot [92].
- The kidnapped robot problem: This kind of situation arises when the robot gets picked up and travelled to another point on the map without the robot's knowledge. The problem can be solved similarly to the global localization problem only if the robot itself knows that it has been kidnapped. The complexity of the problem arises when the robot thinks it knows its position, but in fact, it doesn't [93].

A mobile robot uses on-board sensors to learn about its environment. But often these sensor readings cannot be trusted which means that localization is such a difficult problem for two reasons [8]. The first is sensor noise where sensor readings are dropped over time. This might be solved by taking multiple sensor readings and using sensor fusion

methods. The second problem that might be faced is called sensor aliasing which is a problem that humans rarely encounter but is common for robots [94], [95]. Figure 11 represents the basic working diagram of localization methods. There are also other types of localization methods such as:
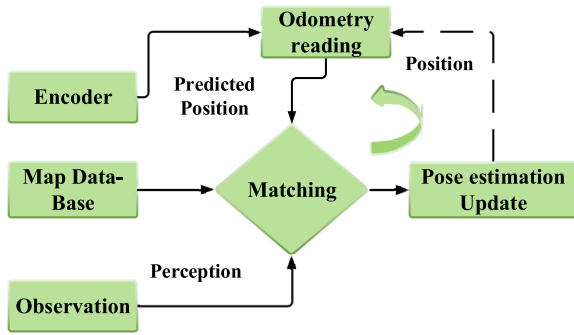

FIGURE 11. Basic Block Diagram of Localization [8].

1) Positioning beacon based localization: By designing active beacon systems and deploying them in the target environment, good localization can be achieved based on wireless communication [96], [97].
2) Light-based Localization: Estimating poses of the robot in indoor conditions based on visible light is a newly emerging field in the mobile robotics field [98].
3) Landmark-based Localization: These techniques use natural as well as artificial landmarks and relative position techniques such as dead-reckoning, and triangular methods to localize the robot [12], [99], [100].
4) Odor based Localization: This type of technique analyses chemicals of the environment and makes decisions about its surroundings. Although this is mainly applicable for harsh conditions or underwater robots, this can also be applied to indoor mobile robots in certain cases [101].

The error from the sensor reading can be minimized by the following algorithms: [102]:

1) Kalman filter localization
2) Markov localization
3) Monte-carlo localization

1) KALMAN FILTER LOCALIZATION
The Kalman filter is well known in the world of dynamic control systems that use the probabilistic algorithm for estimating the state of a time-dependent process and reducing the noise as shown in Figure 12 [103]. It estimates the current state by comparing the previous estate with recent measurements [104].

Consider a linear dynamic system as: [90],

$$X_{k_r+1} = F_{k_r}X_{k_r} + G_{k_r}U_{k_r} + V_{k_r}$$
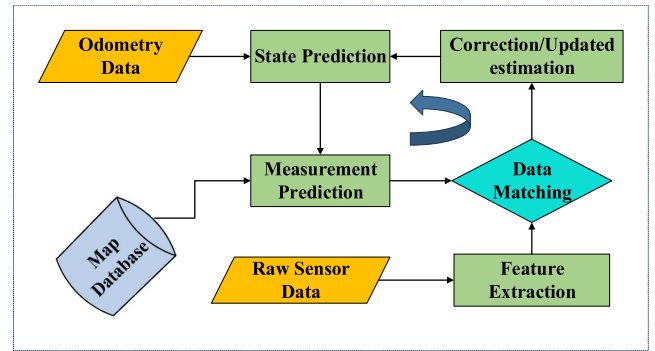$$Y_{k_r} = H_{k_r}X_{k_r} + W_{k_r} \qquad (3)$$

FIGURE 12. Basic Block Diagram of Kalman Filter Localization [90].

where, X_{k_r} is a state vector, U_{k_r} and Y_{k_r} are the input and output vector, F_{k_r}, G_{k_r}, H_{k_r} are system matrices and V_{k_r}, W_{k_r} are Gaussian noise with co-variance matrices such as Q_{k_r}, R_{k_r}.

The estimation of the next states can be represented as [90], [105]:

$$X_{k_r+1|k_r} = F_{kr}X_{k_r|k_r} + G_{k_r}U_{k_r}$$
$$P_{k_r+1|k_r} = F_{kr}P_{k_r|k_r}F_{kr}^T + Q_{kr} \qquad (4)$$

Then, the estimated states are corrected by using a correction step:

$$S = H_{kr}P_{k_r+1|k_r}H_{k_r}^T + R$$
$$W_{kr} = P_{k_r+1|k_r}H_{k_r}^T S^{-1} \qquad (5)$$

Then the finally updated states of the Kalman filter are:

$$X_{k_r+1|k_r+1} = X_{k_r+1|k_r} + W_{kr}V_{kr}$$
$$P_{k_r+1|k_r+1} = P_{k_r+1|k_r} + W_{kr}SW_{kr}^T \qquad (6)$$

This algorithm has poor performance with nonlinear systems that is its main limitation [106].

2) MARKOV LOCALIZATION
It is impossible for a robot to measure its pose directly, even with GPS. The best a robot can do is guess about its state. The best guess for a robot's pose is known as belief. In probabilistic robotics, belief can be represented as conditional probability distributions and can be denoted by $bel(x_{rt})$ [8].

In Markov localization, the credence is described by a probability distribution over all possible states, and the Bayes theorem and convolution operations are used to update the opinion after each sensor reading [107]. It first distributes the configuration position of the robot [x y θ] into a finite number of discrete grid cells where each grid tells possible robot positions [108].

According to Markov's assumption, the only state in the environment which affects sensor readings is the robot's current location, not all the other previous states [8]. The belief is computed by taking the odometry data into account:

$$\overline{bel(x_{rt})} = \int p(x_{rt}|u_{rt}, x_{rt-1}bel(x_{rt-1}))dx_{rt-1} \qquad (7)$$

TABLE 4. Comparison of Localization Techniques.

Method	Ref	Filter type	Distribution	Assumptions	Steps	Application	Problems	Improved version
Kalman filter localization	[122], [123]	Kalman filter	Gaussian/ Normal distribution, Parametric	(i) World is Gaussian (ii) System is linear	(i) Prediction (ii) Correction (ii) Update	Local Pose estimation	(i) Does not work for non-linear systems (ii) Cannot solve global pose and robot kidnapping problem (iii) Cannot handle multi model robot distribution	EKF, UKF [124], [125]
Grid-based Markov localization	[108], [126]– [128]	Histogram filter	Discrete, Non-parametric	(i) The configuration space is a set of discrete cells or histograms (ii) The next state of the robot is predicted only using the current state of the robot but not the previous ones	(i) Prediction (ii) Measurement	(i) Local pose estimation (ii) Global pose estimation (iii) Can tackle robot kidnapping problem (static environment)	Computationally expensive to do for real-time operation	Markov-kalman localization [129]
Monte-Carlo localization	[110], [111], [130], [131]	Particle filter	Arbitrary, Non-parametric	The possible states of a robot in a single moment can be described by particles. Each particle describes a possible state.	(i) Proposal (ii) Measurement (iii) Resampling	(i) Local pose estimation (ii) Global pose estimation (iii) Solves robot kidnapping problem (static environment)	Computationally expensive hence has to be a tradeoff between speed and accuracy	AMCL [132], [133]

where, $bel(x_t)$ is the possibilities of belief which are further updated a:

$$bel(x_{rt}) = \eta p(z_{rt}|x_{rt}, M)\overline{bel(x_{rt})} \qquad (8)$$

Here z_t are sensor readings, and $p(z_t|x_t, M)$ is a probabilistic measurement model [8]. Markov localization does well to compute global localization and solve the robot kidnapping problem as it sums over all the possible positions a robot can be in at a given time [109].

3) MONTE-CARLO LOCALIZATION

Monte-Carlo localization uses a particle filter algorithm to estimate the pose of the bot [110]. In this algorithm, particles are used to represent a probability density function. Each particle is thought to represent a state of the robot is also given a weight for that possibility [111]. The higher the weight, the higher the possibility of a particle being right. The weight of each particle is computed using the ratio between target and proposed function where each function is an estimate of the robot.

The noise and motion model is applied for each particle by sampling from the density function as:

$$x_t^j \sim p(x_t|x_{t-1}, u_t) \qquad (9)$$

The measurements of the sensors are denoted by z_t and are used to compare the proposed state with the target state to compute the weight for each particle as:

$$w_t^j = \frac{target}{proposal} \sim p(z_t|x_t, m) \qquad (10)$$

where the weight w_t^j basically computes the likelihood of each particle being x_t^i given z_t.

The next step creates a current group of particles that keeps the particles having higher mass values and gets rid of particles with lower values. The Monte Carlo localization works better with more samples, but this means more computational complexities. Table 4 compares the localization techniques and shows their pros and cons.

4) POSE GRAPH OPTIMIZATION

Apart from these three techniques, another method named Pose Graph Optimization (PGO) has been popular in the last decade [112]. This method is mainly used for localization in graph-based slam techniques [113]. While a robot moves through the environment, its trajectory can be defined as a collection of its poses over time and these poses are joined together with edges creating some sort of graph. This is known as a pose graph. Pose graph optimization is used to estimate the accurate poses relative to the best possible constraints in the graph, which eventually is used to compute the trajectory of the robot. The poses are usually measured with various methods such as IMU readings, scan matching, dense point clouds, etc. The problem is usually treated as a non-convex optimization problem, where the solution is the maximum likelihood approximation of the comparative poses. Most state-of-the-art techniques to solve PGO are iterative optimization techniques such as Gradient method [114], [115] or Gauss-Newton method [116], [117] etc. These are generally used to compute local minima. Convex relaxations based solutions have recently been popular to solve the global minima [118]–[120]. PGO has also been used in multi-robot localization [121].

C. SLAM

The robot localization problem is closely related to the problem of map building. Without a map, it is impossible for a robot to localize itself properly, and again, without localizing the robot, it is difficult to comprehend the surrounding environment; thus the mapping process can be inaccurate [77]. So to solve this, the concept of SLAM was developed in the 1990s. The SLAM problem is developing an environmental map and determining the right position to track the robot simultaneously with the noisy sensor data available [134].

The problem can mainly be categorized into two types. "Online SLAM", which deals with only the current pose of the robot while "Full SLAM" tracks the robot path always while building the map [11]. If the state of the robot is X_K, and the distance between the states is U_K, then the SLAM problem can be described as,

$$p(X_{k1:K}, M | Z_{k1:K}, U_{k1:K}) \qquad (11)$$

where M is neighboring ambient and the sensor measurement is Z_K. The online SLAM problem can be denoted by:

$$p(x_k, M | z_k, u_k) \qquad (12)$$

SLAM algorithms can also be classified based on the sensors they use. Lidar SLAM, which uses lidars as primary sensors for the environment perception, and Visual SLAM or V-SLAM, which uses vision sensors or cameras as primary sensors. Although SLAM seems to be the ideal solution for mobile robotics, it takes time to implement because of the high dimensionality and a large number of discrete variables in the parametric space [24].

1) PARADIGMS OF SLAM

Most SLAM algorithms were derived from the three basic paradigms. EKF-SLAM, which emerged on the basis of the extended Kalman filter [135]. This algorithm uses a single dense statute vector in order to estimate the position and the features of its surroundings having any error covariance matrix, which represents uncertainties associated with the estimations based on the online SLAM problem [11].

Particle filter based SLAM algorithm is the second principal paradigm of SLAM. It's usually okay for something like localization but in the case of SLAM where a huge number of features are present, makes the system complex that can be overcome by the fastSLAM approach [136] which always works with an adjusted number of particles as,

$$X_t^{[K_l]}; \mu_{t,0}^{[K_l]} \ldots . \mu_{t,N-1}^{[K_l]}; \Sigma_{t,0}^{[K_l]} \ldots . \Sigma_{t,N-1}^{[K_l]} \qquad (13)$$

where $X_t^{[K_l]}$ is the estimated sample route and estimation of each feature which are described by a *2D Gaussian* which have a N set of mean $\mu_{t,n}^{[K_l]}$ and variance $\Sigma_{t,n}^{[K_l]}$.

The control commands of the motor u_t update the estimated position of each particle as:

$$X_{t+1}^{[K_l]} \approx p(X_{t+1} | X_t^{[K_l]}, u_t) \qquad (14)$$

The next step is to compute the importance weight $w_t^{[K_l]}$ based on the measurement readings z_t as:

$$w_t^{[K_l]} = p(z_t | x_t^{[K_l]}, \mu_{0,N-1}^{[K_l]}, \Sigma_{t,N-1}^{[K_l]}) \qquad (15)$$

fastSLAM has been improved in several ways such as GridSLAM [137], gMapping [138]–[141] etc. are Grid-based versions of fastSLAm.

The third general family of algorithms is called graph-based SLAM. These algorithms treat the SLAM problem as a sparse graph where pose and landmarks of the robot are treated as nodes in the sparse graph [142]. The information of the nodes are portrayed as robot poses $x_{k_1}, x_{k_2}, x_{k_3}, x_{k_4}, \ldots . . x_{k_N}$ and n number of landmarks in the map as $m_{k_1}; m_{k_2}, m_{k_3}, m_{k_4}, \ldots . m_{k_n}$. The constraints between the nodes are thought of as the relative distance between successive nodes, which is influenced by the odometry data u_t. The constraints are relaxed, such as springs to get the best possible estimation of the pose and the map of the robot. The system can be seen as a spring-mass model [143] and solved by computing the minimal energy state of the model. Some of the popular open source graph-based algorithms are g2o [117], GTSAM [144], HOG-Man [145], KinectFusion [146], PTAM [147], TORO [148], CPL-SLAM [149], [150] etc. A comparative analysis of these three paradigms has been shown in Table 5.

2) LIDAR SLAM

The lidar scanners are largely used in SLAM as the primary sensors for indoor mapping and localization in which the time of flight (ToF) method is used to calibrate the readings of the lidar sensor. Larger detection range, lower dependency on the environment and lower cost of acquiring lidar data are the advantages of lidar SLAM [158], [159].

Single-line lidar sensor can be used in 2D applications due to the presence of single pair of transmitter and receiver which can be used in scene-matching and mapping problem [140], [145]. The lower detection range is the main limitation of this lidar which can be overcome by developing multi-line lidar sensors that increase the navigation range by establishing a 3D map. It provides the facility to capture 360 degree field of view with the help of a point cloud.

The algorithms can be categorized into organized [160], [161] and unorganized [162] based on the environment, and the type of point clouds [160]. Another way to deal with 3D point cloud is to perform data association among point clouds and look for closest points, which is called Iterative closest point (ICP) algorithm [163], [164].

3) VISUAL SLAM

In the earlier years of SLAM, research was focused more on 2D SLAM. Over time, 3D maps with more information were needed for the mobile robots to do more complex, specific tasks in dynamic environments since 2D maps do not hold distinctive features of the environment. VSLAM stands for a visual slam, which involves solving the problem where the

TABLE 5. Comparison Table Between Basic Paradigms of SLAM.

Method	Properties	Pre-requisites	Applications	Advantages	Limitations	Open-sourced ver.
EKF-SLAM	Filter based approach, uses extended Kalman filters. Uses single dense state vector for state estimation and an Error co-variance matrix for representing uncertainty.	Number of landmarks, Known data association. The motion model is based on velocity	Active SLAM (can be modified to do Full SLAM) in small environments. Used in airborne, underwater or indoor robotic applications.	(i) Can handle non-linear situations, provides optimal state of the robot through MMSE (Minimum mean-square Error) (ii) The Error co-variance matrix converges easily.	(i) Time complexity in general is quadratic hence computationally expensive for long operating conditions. (ii) Doesn't work with uncertain data association, needs special treatment to converge in such cases. (iii) Cannot do FULL SLAM in normal cases.	[151], [152],
fastSLAM	Filter based approach, uses Rao-Blackwellized particle filters. Uses a sample path and "N" numbers of a 2D gaussian (EKF). Each gaussian has a mean and a variance. The particles are updated after each prediction step using importance weight.	The number of particles "K" in each step has to be determined beforehand.	Used for Active Slam, can do full slam for relatively larger maps than ekf-slam can manage.	(i) Time complexity in general is logarithmic, computationally less expensive than EKF-SLAM. (ii) Can handle multiple data associations, deal with unknown data associations to some extent. (iii) Can do FULL SLAM for medium sized maps.	(i) No defined methodology exists for fixing the number of particles "k" in the initial stage. It has to be done by "educated guessing". (ii) Repeated visits to same mapped areas cause particle depletion, which can prevent the algorithm from building consistent maps.	[153], [137], [154], [155], [156]
Graph-based SLAM	Optimization-based approach. The configuration space is thought of as a sparse graph. Uses optimization techniques such a Bundle adjustment.	The nodes in the sparse graph are considered as landmarks & poses. The edges between the nodes are considered odometry readings.	FULL SLAM. Used for generating large accurate maps.	(i) Memory requirement is linear. (ii) Varieties of efficient optimization techniques exists; hence more popular among researchers. (iii) Suitable for FULL SLAM, can build very large maps and is the preferred method for huge areas. (iv) Recent optimized methods can do online SLAM very efficiently with visual sensors only.	(i) Most of the initial algorithms were offline algorithms. (ii) Though the optimization techniques are efficient, they are expensive. (iii) Data association needs to be optimal for optimum performance.	[117], [145], [114], [157]

primary sensor of the system is a sensor based on vision, such as a camera [165].

Three different types of cameras are used for VSLAM. When using only a single camera, VSLAM is often called monocular SLAM [166]. Another type that uses stereo vision for SLAM systems uses a set of cameras which allows depth to be readily perceived. RGB-D cameras use infrared light to measure the depth of objects. Visual SLAM can be generally classified into two different methods [78]:

a: FEATURE-BASED METHOD
This method has been the most commonly used method in SLAM. Images are the type of data that hold a high volume of information. Thus it was necessary to pull out distinct features from these images for getting useful features only which can identify particular areas using techniques such as SIFT [167], SURF [168], ORB [169] etc. Feature-based methods can be categorized into two types of approaches: filter-based approaches and key-frame based approaches. Filter-based approaches have been used for monocular slam [170] using EKF and fastSLAM to tackle the non-linearity issue. Key-frame based approaches such as PTAM solve the localization and mapping task in parallel.

b: DIRECT METHOD
The direct method uses the pixel information directly from each frame to track the pose of the robot and perform mapping. Although feature-based methods are quite robust, they tend to perform poorly in low texture scenes as well as when scenes are repeated as the feature extraction processes are texture sensitive [78]. Direct methods use pixel to pixel brightness information and try to estimate the pose by optimizing the iterative process and reducing the initial photometric errors [11].

Some of the more popular direct methods such as DSO [171], LSD-SLAM [87], DTAM [172] used monocular vision as their primary sensor. Some used RGB-D sensors as the primary sensor such as Kinectfusion [173], Kintinuous [174], BAD-SLAM [175] etc.

VSLAM is still very much an ongoing research topic, including incorporating the power of modern GPUs, to create a more dense map with semantic information and better tracking [78], [176], [177]], [178], [179].

D. PATH PLANNING
For a particular environment, a mobile robot determines an optimal or sub-optimal collision-free route towards its desired

goal from the initial location according to a specific performance criterion. Path planning techniques are divided based on the environment [13] in which objects may be fixed or movable. Path planning techniques can also be divided into two divisions: global path planning and local path planning.

Global path planning is concerned with techniques that require every detail about the surroundings before the robot starts navigating such as, Dijkstra algorithm, [180], A* algorithm [181], Visibility graph [182], Artificial potential field method [183], and cell decomposition method, etc.

In local path planning, the surrounding environment is usually unknown or barely known in some cases [184]. The information from the surrounding environment is generally extracted from the robot's onboard sensors. Some popular algorithms are Artificial Neural Networks [185], Fuzzy logic [186], Neuro-Fuzzy logic [187], PSO [188], GA [189], and ACO [190], RRT [191].

1) PRINCIPLES OF PATH PLANNING

Figure 13 shows a basic diagram of the path planning process which follows three steps: (i) environmental modelling, (ii) optimization criteria, and (iii) path searching algorithm [18].

Environmental Modeling is the step which creates a description of the robot's surroundings, which is understandable by computers. Environmental modeling dramatically reduces the planning complexity and computational power, as well as navigation time.

Optimization Criteria is used to model a path planning algorithm in the following manner [9]:

(i) Minimum time needed, which controls the productivity of the robot.

(ii) Minimum jerking, which ensures the exactitude, proficiency of work, and proper maintenance of the equipment.

(iii) Least power or least actuator exertion needed in order to save energy.

(iv) Hybrid criteria, such as minimal time and power.

(v) Minimum Path length: The path length D_m is defined as [192], [193]:

$$D_m = \sum_{j=0}^{n-1} \sqrt{(x_{j+1} - x_i)^2 + (y_{j+1} - y_j)^2} \qquad (16)$$

where, x_j is the value of X coordinate axis and y_j is the value of Y coordinate axis.

(vi) Smoothness: Smoothness is defined as [193]:

$$S = \alpha(1 - \frac{DA_l}{N_f - 1}) + \beta(1 - \frac{s_{min}}{N_f}) \qquad (17)$$

where α & β are weighted coefficients, DA_l is the angle of deflection, N_f is the number of path segments, and S_min is the number of minimum path segments in the path [18].

Path Searching Algorithm is adopted to find a collision-free route between the onset point and the destination point in the state space, which should satisfy a set of optimization measures including route length, evenness,
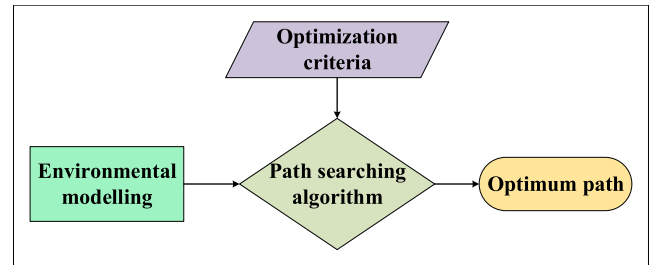

FIGURE 13. Basic Working Process of Path Planning Algorithm [18].

safety standard, etc. Path planning algorithms can be classified as (i) Classical Algorithms, (ii) Heuristic Algorithms and (iii) Reactive algorithms.

2) CLASSICAL ALGORITHMS

Classical algorithms rely on two states. These methods decide whether there is an optimum path or not. The main limitation of the classical approach is that these methods need more computational power, and they cannot adapt to uncertain conditions. Some of the best known classical methods are Cell Decomposition (CD), Potential Field (PF), Road Map. etc [194].

a: CELL DECOMPOSITION METHOD

The cell decomposition method divides the robot's surrounding environment into grids or cells of two kinds. One kind is a free cell, in which no object is present, and the other has items in it. These methods are generally used to construct graphs which can be later used with graph-based search algorithms for optimal path planning. The basic algorithm structure is as follows; [18], [194]:

- Divide the search region into connected simple regions called a cell.
- Determine the cells that have no object as open cells
- Identify the open cells which are closed to each other and construct a ''connectivity graph''
- Find the cells which have the initial and final goal point and determine a collision free path in the connectivity graph from the start cell to the final cell.

During traveling, open cells are considered to achieve path planning, extending from the primary and final (goal) points. Closed cells (cells that contain obstacles) adjacent to the path are further apportioned into two sections to attain an open cell. Figure 14 shows how cell decomposition methods model an environment. The cell decomposition method has the problem of requiring a lot of memory to analyze the surrounding environment.

b: ARTIFICIAL POTENTIAL FIELD

The potential of the workspace is described and the robot calculates a repulsive force from field activity in this method. The goal point will act as an attractive force for the robot and the potential function can be described as [195], [196]:

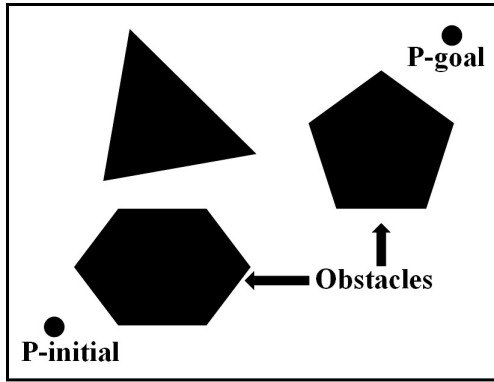$$U(x) = U_{att}(x) + U_{rep}(x) \qquad (18)$$

FIGURE 14. Cell Decomposition Method [8].

where, U_{att} is the attractive potential and U_{rep} is the repulsive potential. U_{att} can be written as:

$$U_{att}(x) = \frac{\xi \rho^2(x.x_{goal})}{2} \quad (19)$$

where ξ is the positive gain factor and x, x_{goal} is the Euclidian distance between the robot and the target destination. Thus the attractive force can be represented as:

$$F_{att} = -\nabla U_{att}(x) = \xi(x_{goal} - x) \quad (20)$$

The repulsive potential is possibly be denoted by the formula:

$$U_{rep}(x) = \begin{cases} \frac{\eta}{2}.(\frac{1}{\rho.x.x_{obs}} - \frac{1}{\rho_o})^2, & \text{if } \rho.x.x_{obs} < \rho_o \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

where η is a scaling factor. Thus, the repulsive force can be given as:

$$\begin{aligned} F_{rep} &= -\nabla U_{att}(x) \\ &= \begin{cases} \eta.(\frac{1}{\rho.x.x_{obs} - \rho_o}).\frac{1}{\rho(x)^2}.\frac{x - x_{obs}}{\rho(x)^2}, & \text{if } \rho.x.x_{obs} < \rho_o \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (22)$$

The artificial potential field algorithm has a number of issues:

- There might be a local minimum depending on the dimension and configuration of the objects around the robot points
- Another problem arises for objects that are of concave shape which causes more than one minimal distance to arise creating oscillations on the resultant paths

c: PROBABILISTIC ROAD MAP APPROACH

The probabilistic road map (PRM) method relies on the arrangement of space which can be represented as a set of all configurations the robot can obtain which is used to identify the collision function. Then, sampling of a grid in the configuration space is undertaken by checking whether or not the grid point is in free space using the collision check function. All the grid points in free space are treated as nodes in the graph. The nodes are then connected by lines which are called edges [197].

The PRM method chooses the points in the configuration space randomly and tests whether the point is in free space using the collision check function. If the node is in free space, then it tries to see if it can connect with the closest existing node and create a path. The boundaries between the random samples can be thought of as road-maps that form a network that hopefully spans the free space in the configuration space [198]. Two popular road map approaches are the Visibility graph and Voronoi graph.

A visibility graph is a diagram where the vertices of all polygonal obstacles are thought of as nodes for the sampling free space along with the initial and final goal node. All the lines connecting the vertices and the initial and final node are identified as edges of the road-map. So the roads are very close to the obstacles; hence the resulting path in almost all cases would be a minimum-length solution [8]. This method is very slow and inefficient in environments with many objects.

The Voronoi graph is the opposite of a visibility graph in the sense that it tries to stay as distant as possible from the obstructions. It describes a summation of equidistant points from the next two or more obstacles, containing the configuration space [199]. Thus it ensures maximization of the interval within the bot and the obstacle. Its primary disadvantage is that localization sensors with a short-range might fail to detect the obstacles as it keeps maximum distance from obstacles. Figure 15 shows how PRM methods connect the shortest paths together.

3) GRAPH-BASED SEARCHING ALGORITHMS

These are known as the most straightforward methods for finding a path for the robot which needs to know the environment and the robot's primary and terminal position beforehand for working purposes. Some of the most notable heuristic algorithms are [13]: Dijkstra's algorithm, A*, D*, etc.

a: DIJKSTRA'S ALGORITHM

This algorithm is applied to find the shortest route out of several possible paths in a graph by dividing it into a grid cell [200]. The algorithm focuses on two components; nodes and weighted edges. The algorithms work by calculating edge weights between two nodes and calculates the minimum value as it gets closer to its target node.

b: A* SEARCH ALGORITHM

One of the major problems of Dijkstra's algorithm is while it pursues the route that seemed the shortest currently, it isn't concerned about its direction of approach. To solve this, the A* algorithm has been proposed based on Dijikstra's algorithm with some modifications which also accounts for the Euclidian space [200]. The algorithm uses the evaluation function: $f(n_w)$.

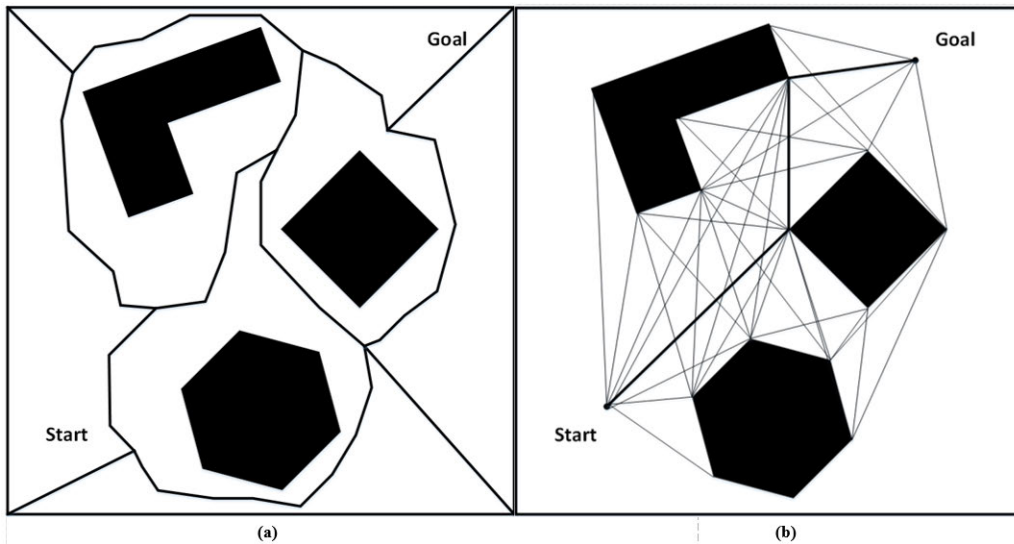$$f(n_w) = g(n_w) + h(n_w) \quad (23)$$

FIGURE 15. Artificial Roadmap Approaches; (a) Voronai Graph and (b) Visibility Graph [8].

where, $g(n_w)$ is the actual cost and $h(n_w)$ is the heuristic function. This algorithm has much higher efficiency than Dijkstra's algorithm.

c: D* SEARCH ALGORITHM

Although the A* algorithm works well in a static environment, its performance drops in dynamic environments where objects may change with time. To tackle this, the D* search algorithm was introduced [201]. It is capable of planing real time-optimal path. It works similar to A*, but while with A*, the edge weights are constant, with D*, it is ever-changing. So as A* discards the information about the nodes it has visited, D* keeps that information to plan and re-plan efficiently [202].

4) EVOLUTIONARY ALGORITHMS

Evolutionary algorithms work really well when the environmental model is unavailable or unrealistic, and dynamic. These algorithms are inspired by stimulus-response, which is the biological reaction to any action. Reactive planners including fuzzy logic, particle swarm optimization, neural network, artificial bee colony, ant colony optimization, genetic algorithm, shuffled frog leaping algorithm and several miscellaneous algorithms have been used researched.

a: GENETIC ALGORITHM

The genetic algorithm (GA) solves the problem of path planning by addressing the issue as a chromosome. The possible solutions are addressed as populations. The mechanism of the GA is moving from one community of chromosomes to another using an operation called "Selection." There is a fitness evaluation function that evaluates and ranks all the communities [203], [204].

b: ANT COLONY OPTIMIZATION ALGORITHM

This algorithm has been modeled after the ant's movement in search of food. While searching for food, ants release
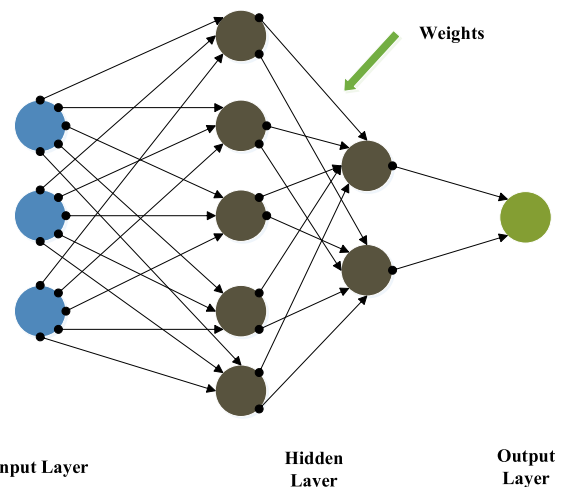

FIGURE 16. Basic Architecture of Artificial Neural Network [209].

secretion after short intervals leaving it as a reference so that other ants can follow the path. This secretion is known as a pheromone. The more pheromone is there in a path, the more optimized that way is. Only one path will have a pheromone after the optimized path is found [205]–[207].

c: ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANN) follow the operation of the neuron of a human brain consisting of input, output, and hidden layers with weights. These weights are randomly initialized at the beginning but change accordingly to get optimum results. The connections also have activation functions with the weights. With simple weights activation functions, ANN can learn almost anything with the right amount of training data [208]. Figure 16 shows a basic neural network with two hidden layers, along with the input and output layers.

d: PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) follows a familiar pattern with a bunch of particles in which individual scrap presents a possible accomplishment. It is employed nowadays as a solution to the navigation problem of robots. Mohemmed *et al.* [210] proposed a PSO-based algorithm that finds the shortest path from A to B. Zhang *et al.* [211] developed a multi-destination PSO to make the robots navigate in uncertain and dangerous environments where many hazards may lie. Figure 17 shows a basic working diagram of the particle swarm optimization algorithm.

e: RAPIDLY-EXPLAINED RANDOM TREE ALGORITHMS

Modern autonomous indoor robots are developed to perform some more tasks related to interact with the external environment. This robot and environmental interaction have various styles and types such as the systems where reconstruction of shapes of any three-dimensional object [212] and also a navigation system based on vision, which is mostly used in drones [213].

The aforementioned tasks can be easily done by employing a rapidly-explained random tree (RRT) approach which is designed to administrate the nonholonomic constraints and a higher degree of freedom [215]. In [215], the algorithm was successfully applied to all kinodynamic, holonomic, and non-holonomic planning problems with degrees of freedom up to twelve. Table 6 highlights different path planning approaches.

E. OBSTACLE AVOIDANCE

Obstacle avoidance mainly refers to the change of trajectory of the robot, which allows navigation in some unknown environment by avoiding collisions. In various researches, obstacles have been avoided in different methods, including in [252], a method is proposed based on the artificial potential fields principle, which has been extended in [253] with the addition of schema motors, and a method for swarm robots to avoid obstacles is described in [254] which is based on the self-organizing migration method.

If the robot cannot estimate the presence of, and avoid, any obstacle in its trajectory, it will not accomplish its end position which obviates the main purpose of a mobile robot. Basically, this feature can save the robot from any kind of crash which makes the movement less restrictive or fragile. The resultant gait contains both the function of recent or current reading of the robot sensors and also the goal position or its relative location to the goal position. An indoor mobile robot comprises various functions, of which avoiding obstacles is a vital part, which is a must to keep overall monitoring of the architecture for maneuvering tasks [256].

There are also some systems where the mapping is integrated with obstacle avoidance, where during the robot's travel toward its goal the sensor data sampled continuously along with building the map with an immediate update and the algorithms for obstacle avoidance continue the work of mapping-information simultaneously [257].
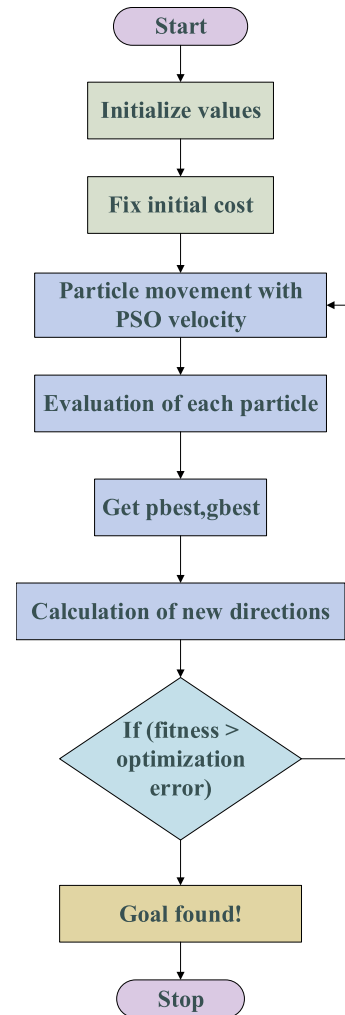

FIGURE 17. Basic Block Diagram of Particle Swarm Optimization Algorithm [214].

Some algorithms presented here regarding obstacle avoidance depending on varying degrees in the global map existence and the appropriate information of the robot about its relative position towards the described map.

1) BUG ALGORITHM

The bug algorithm is considered as one of the earliest of all algorithms [258], [259] which can enable the navigation of the whole periphery of the encountered obstacle. It also allows the robot to decide the most suitable way to move toward the goal [12]. The robot can compute the new path [260] from its departure point (x_1, y_1) to the destination point (x_2, y_2) using two equations to find the slope m and y-intercept c.

$$m = \arctan \frac{y_2 - y_1}{x_2 - x_1} \quad (24)$$

$$C = y_1 - cx_1 \quad (25)$$

This is an inefficient approach but it ensures that the robot reaches the goal area. However, it only leaves when it can move in the direction of the goal, as the trajectories are shown in Figure 18.

TABLE 6. Summary of Path Planning Algorithms.

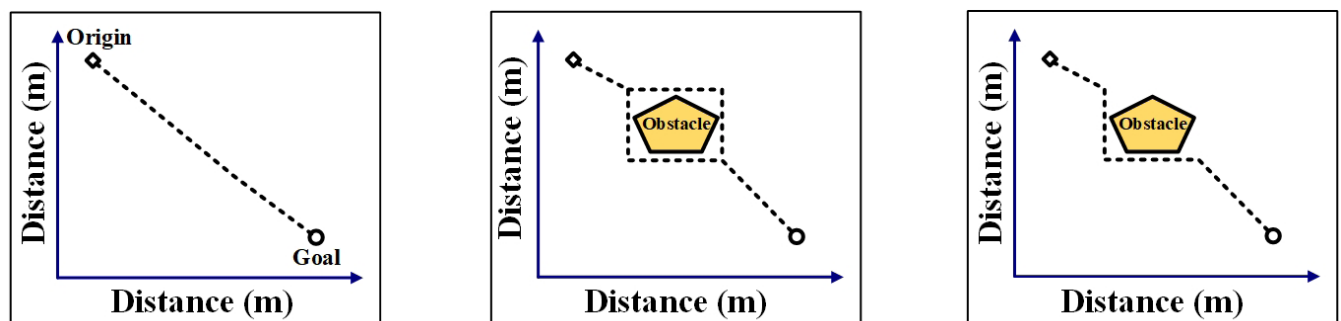| Algorithm type	Method	Ref.	Intuition	Application	Recent developments
Classical Algorithm	Cell decomposition method	[216], [217], [218]	Divides the environment into occupied space and free space	Mainly used for core environment modelling	[219]
	Artificial Potential field approach	[195], [196], [220], [8]	Models the environment using potential field theory and uses elements of environment as positive and negative potentials to do path planning	Global and local path planning	[221], [222]
	Probabilistic Roadmap approach	[197], [223], [199], [224], [225], [226], [227]	Assumes the environment as a configuration space and uses obstacle and a sampling-based algorithm to determine the path	Global and local path planning	[228], [229]
Graph-based search algorithms	Dijkstra's algorithm	[230]	Used to search the quickest route out of a number of possible paths in a graph	Global path planning	D* extra lite [231], HCTNav [232], nafisnav [233]
	A* search algorithm	[200]	Improved version of Dijkstra's algorithm with heuristics which is capable of tracking its direction	Global path planning	
	D* search algorithm	[201]	Improved version of A* with dynamic path planning capabilities.	Global and local path planning	
Evolutionary algorithms	Genetic Algorithm	[234], [235], [235], [236], [237], [238]	Applies the process of evolution in biological bodies to the robot path planning problem.	Dynamic path planning	neuro-fuzzy [239], Deep-learning [240], Shuffled frog Leaping [241], PSO [242], ACO [243], RRT [244], [245], [246]
	Ant Colony Optimization	[205], [206], [207]	Motivated by how ants find optimized paths through their pheromone	Dynamic path planning	
	Artificial neural networks	[247], [248]	Creates intelligent path planning models inspired by the backbone of biological brains, neurons.	Global and local path planning	
	Particle swarm optimization	[249], [210], [211]	Models the social-behaviour of clusters of animals such as bird-flocks or fish-schools. By mimicking behaviours of these social animal, the algorithm optimizes any search problem.	Dynamic path planning	
	RRT	[215], [250], [251]	special kind of graph called a tree.	Dynamic path planning and obstacle avoidance	

FIGURE 18. Trajectories of Bug Algorithm [255].

IBA [255] is the improved Bug algorithm but it can also be inefficient in certain situations. There are many more Bug algorithms such as Tangent Bug [261], I-Bug [262] etc.

2) THE VECTOR FIELD ALGORITHM
The Vector Field Algorithm is basically an updated version of the virtual force field (VFF) method [263]. The VFF method

employs only a single step technique but VFA works with a two stage data reduction technique. While piloting the robot simultaneously in the direction of the target, these algorithms allow the detection of any unknown obstacles and also allow the robot to avoid the collision [12].

The cost function is [264], G = a. target_direction + b. wheel_orientation + c. previous_direction

This real time obstacle avoidance method can control vehicles to allow them to enter any slender passage such as narrow corridors without compromising the speed or colliding with any unexpected obstacles even in a very densely cluttered place [265].

3) THE BUBBLE BAND TECHNIQUE
The concept of a bubble is mainly the highest subset (local) of free place around the configuration given to the robot where it can move in any direction and travel without any collision. The bubble can be created with the help of a simplified model along with a range of information that are available in the map of the robot [266].

With the help of a simplified model of the geometry of robots, the actual shape of the robots can also be taken into account while calculating the size of the bubble. This widely adopted idea, which is considered an extension of vehicles which are nonholonomic of the elastic band concept [267].

4) THE CURVATURE VELOCITY TECHNIQUE
The Curvature Velocity Technique prefers operating inside the velocity space rather than in the Cartesian or configuration space of the robot [268]. In addition, the commands of this method are chosen by the maximization of objective functions which increase the vehicle navigation, and obstacle avoidance properties such as the speed, direction of movement towards the goal and also safety. This technique consists of two methods.

a: THE BASIC CURVATURE VELOCITY APPROACH
This basic method [268] develops the full kinematic and also some dynamic constraints, which are taken into account to avoid obstacles. This is considered as an advantage, adding the physical constraints both from robots and the environment towards the velocity space, consisting of the translational and also rotary velocity. For this, the robot is assumed to be travelling along arcs of circles only.

b: THE LANE CURVATURE METHOD
The Lane Curvature method (LCM) is an improved approach than the previous one which is based on the CVM, but here the problem with guiding a robot through the corridor's intersections [269]. This upgraded method has been able to provide better results in various experiments than the previous CVM one.

5) THE DYNAMIC WINDOW APPROACH
Another common method for robot kinetic constraints is the dynamic window method for obstacle avoidance [270]. Two such approaches are mentioned below.

a: THE LOCAL DYNAMIC WINDOW APPROACH
In Local Dynamic Window Approach [270], the robot kinematics are considered with the help of searching a fine velocity space, which is set to be an angular velocity. In this method it is considered that the robots only move in some circular arcs, during one time period at least, to represent each tuple. There an objective function [8] can be described preferring three criteria; the fast forward motion, the alignment to goal heading and also maintaining large distance to obstacles. Figure 19 shows the basic working procedure of dynamic window procedure. The objective function,

$$O = x.heading(m,k) + y.velocity(m,k) + z.distance(m,k) \tag{26}$$
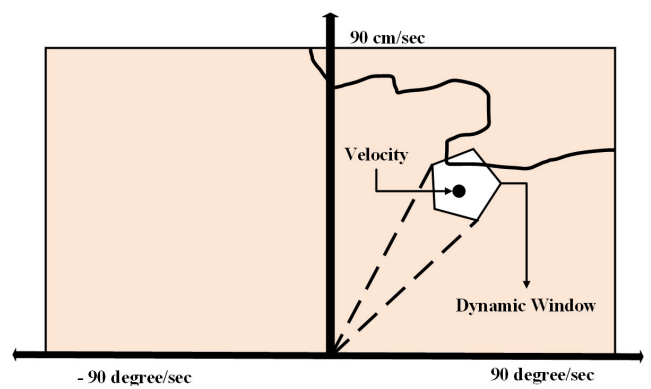

FIGURE 19. Dynamic Window approach [270].

Here, *heading* is expressed as a measurement of progress in the line of the location which is our goal, *velocity* is the robot's forward velocity (fast movements encouraged) and *distance* is the distance towards the obstacle closest to the trajectory.

b: THE GLOBAL DYNAMIC WINDOW APPROACH
This advanced method comes along with the universal conception addition of the local approach. It can solve several problems with definite global thinking, real time and also at high speed minimal free obstacle avoidance, increasing the cycle frequency, and robot speed.

6) THE NEW HYBRID NAVIGATION ALGORITHM (NHNA)
Mainly based on two layers, one deliberative and another reactive layer, the New Hybrid Algorithm focuses on the navigation and obstacle avoidance process, where both the layers work independently to each other [271], [272]. In this algorithm, the deliberative layer can plan a reference path based on storing the previous information with the help of the A* search algorithm. On the other hand, the reactive layer is an independent layer steering robot towards the planned track by the deliberative layer.

7) VISION SENSOR-BASED ALGORITHM
Vision sensor-based algorithms are the highest efficiency algorithms that are able to detect obstacles and also identify the regions which are obstacle free. Some of the vital

TABLE 7. Overview of Different Algorithms Used for Obstacle Avoidance.

Method Name	ref	Sensors Used	Robots type	Efficiency	Observations
Schegel's Method	[274]	Laser Scanner (360°)	Forklift Tricycle, Circular Synchro drive	High but takes up more memory.	This method permits the change in shape apart from circular.
Nearness Navigation Diagram	[275], [279]	Laser Scanner (180°)	Circular Holonomic	Higher than VFH method.	This method ban deal with unknown, unstructured and dynamic environment
The Gradient Method	[280]	Distance Sensor (180°)	Approximately circular Non holonomic	High	This grid global path planning method works on closeness to obstacles to generate continuous interpolations.
Bug Algorithm	[258] , [259]	Distance sensors like IR sensors, tactile sensors, sonar sensors		Low. May bring the robot way out of the goal.	Easy to operate but accuracy level is not even close to acceptance rate. Selects paths that are very long.
Vector Field Histogram (VFH)	[263], [281], [282]	Sonar Sensors	Non holonomic (GuideCane) And also Synchro-drive (Hexagonal)	High in calculation accuracy but consumes more power, processor and also memory	High computations are required ending up eating more memory also difficult for micro-controllers.
The Potential Field method	[283]	Distance sensors like IR sensors, sonar sensors.		Low in accuracy in calculations.	Selects shorter path so required time is less.
The Follow the Gape Method	[284]	Ultrasonic sensors, lidar Sensors, camera optical velocity sensor		Very high. Able to find shortest path toward the goal.	Unable to work for objects of 'U' shaped.
The Vision Sensor Based Method	[285]	Kinect camera sensor, Sonar sensor.		Very high Depending on the equipment used calculations are quite accurate.	Consumes more time for calculation. Best for larger use rather than mini robots with micro-controllers and also requires higher processors and specific software such as MATLAB
The New Hybrid Navigation Algorithm	[286]	Laser sensors, Sonar sensors		Medium But more efficient than the Bug and Hybrid algorithms.	Calculations takes more time for better accuracy.

achievements unlocked by this method can be figured out in three basic steps [273]:

1) A new learning scheme for the monocular disparity estimation
2) The use of pixel classification with a view to finding the obstacle inequality estimation
3) And finally, generation of regions which are obstacle free

These regions, which are free from all the obstacles are applied in drawing a zone which is free from collision to find the safest path for the robot.

8) OTHER APPROACHES

The methods described above are some of the most popular ones. There are some more that are vastly used and have more complex designs but can solve the obstacle avoidance problems more precisely along with others. These are The obstacle avoidance approach by Schlegel [274], Nearness diagram for navigation [275], the Gradient method, the method by Tzafestas & Tzafestas [276], which is an integrated mapping and obstacle avoiding method, the method by Chen and Quinn, which is inspired by nature [277], a novel obstacle avoidance algorithm ''Follow the Gap Method'' [278] and so on. Table 7 consists of an overview of some of these techniques.

VI. INTEGRATION OF DEEP LEARNING IN MOBILE ROBOTS

Deep learning leverages the power of neural networks to make rules for observable things, making decisions based on observations without developing rules for a task, which is normally done to develop an algorithm for a specific task. Deep learning or so-called learning-based methods worked because of the massive amount of data that started generating because of the internet and the computational power is powerful enough to handle the big data. Three key advantages to using deep learning methods in mobile robot systems are [19]:

- Deep learning-based methods can generate universal models that can automatically extract features from the

data without much mathematical modeling for different conditions such as abnormal lighting, blur motion, or cases where the camera needs calibration by hand, which is hard to do [287]. These models are able to match in some ways human-level reasoning and apply it in methods such as SLAM to create semantic maps [288], [289], or taking visual odometry to the next level [290], [291].

- These methods have the ability to learn from past experiences and improve their accuracy and make decisions in dynamic scenarios. These abilities can be leveraged to solve some challenging problems in robotics such as navigation [292], ego-motion detection from unlabelled videos [293], and decision making in dynamic environments [294].

- Probably the most crucial aspect of deep learning is that it can leverage the increasing power of our computers and the amount of sensor data that can be used in the current world. The growing number of possible sensors we can use to perceive the environment is increasing day by day. To get close to human-level perception, combining these sensor data seems to be the way to go. Recent advances in processing power mean that parallel computing can process a large amount of data efficiently, and deep learning models thrive with big data, giving better performance than traditional algorithms. Hence deep learning-based solutions seem to be the way to go forward.

Deep learning-based methods have been rapidly researched in the sector of robotics in the last decade. For autonomous mobile robots, some of the core difficulties, for instance, odometry estimation, SLAM, navigation, etc., have seen a more cognitive performance; thanks to deep learning-based approaches.

A. ODOMETRY
Deep learning has enhanced visual odometry (VO) that local position and orientation with only vision sensors and also have improved sensor fusion methods such as visual-inertial odometry (VIO) as well. While normal VO methods relied on hand-coded features to work, which is very time consuming and can vary for different rooms altogether, DL based methods are able to extract rich features from video frames, automatically taking VO to another level. These methods can be classified into End-to-End VO and Hybrid VO. End-to-End VO uses only deep neural networks to do supervised or unsupervised learning to solve the VO problem. Supervised VO is trained on labeled images extracted from video frames to capture the transformation from frames to motions. Some earlier works are [295], [296] although they failed to achieve end to end learning. DeepVO [290] was able to achieve end-to-end learning with a combination of a CNN and an RNN to extract features from video frames and map them to robot motion. It outperformed the likes of VISO2 [297], ORBSLAM [298] in KTTI dataset [299]. DeepVO was further improved by [300]–[302] for better generalization and memory

function to upgrade the accuracy of pose estimation. Unsupervised methods have also been of great interest with works such as [293], [303]–[305]. However, these still perform well below supervised methods and still have problems. VIO models uses sensor fusion techniques for perception using inertial sensors like IMU and vision sensors to reduce sensor noise for better pose estimation [306]–[309]. Hybrid methods combining classical models and deep learning approaches have seen a higher accuracy in terms of performance. Some hybrid algorithms like D3VO [291]) even outperform VIO algorithms such as DeepVIO [309], VIOLearner [307] as well as classical VO systems such as DSO [310], ORB SLAM [298] in KTTI benchmarks.

B. SLAM
Deep learning has also been applied to SLAM, another key element of mobile robotics. Classical SLAM systems today are quite robust and solved to some extent, but several key problems still have attracted the deep-learning community's attention. Bundle adjustment, accumulative error are some of the optimization problems related to localization in SLAM. LS-Net [311] tried to solve the Bundle adjustment problem by using deep neural networks to learn analytical parameters needed to satisfy local constraints for the pixel to motion transformation. BA-Net [312] integrated the Levenberg-Marquardt algorithm [313] inside a CNN towards achieving end-to-end learning for solving the bundle adjustment problem. A major problem for mobile robots has been the accumulative error, which occurs when a robot is operated for a prolonged time. As DNN is very robust when it comes to extracting rich features from a scene and predicting pose, it was evident for practitioners to use DNN to reduce accumulative errors. These DNN algorithms have been used in the back end of an established SLAM system in order to mitigate the accumulative errors over time while traversing. Some recent examples would be CNN-SLAM [314] which works on the back-end of LSD-SLAM [87], DeepTAM [315] which feeds both pose and depth information extracted through a DNN to a classical system DTAM [172]. These learning systems optimize the overall global pose of the robot. Another common problem throughout the years for the SLAM community has been loop closure. This is a unique problem and has been much harder to solve, using only lidar-based SLAM solutions. Classical systems used bag-of-words [316] to store features of already seen geometry in order to use them later on using handcrafted features to solve the loop closure problem. The process is very complex and often fails in adverse conditions such as a change in light intensity, weather, or dynamic objects. Hence CNN has been used in some models to recognize high visual features for resolving the loop closure problem. One earlier work on this is [317]. Later, CNN words [318] was introduced, which uses CNN to extract the features on a higher level and incorporate BoW in it to get better accuracy in reducing loop closure. Some deep auto-encoder based solutions were proposed, which can encode complex features and stores them automatically [319]–[321].

Very recently, OverlapNet [164] was proposed, which overlaps scans from two similar 3d lidars and training a DNN using this data, which can be integrated with an existing lidar-only SLAM system to eradicate loop closure.

C. NAVIGATION

Safe and successful navigation is the primary goal of a mobile robot. Though standard navigation in terms of a mobile robot hovering around a static environment and getting from point A to point B is considered solved, the real-world scenario is not so easy. Reinforcement learning has received much attention in order to achieve human-level awareness while traversing through a densely populated dynamic environment. It is an algorithm that mimics the human learning process, which works by repeating the trial and error process until the objective is completed. The reward system drives this process as the system tries to achieve as maximum reward as possible. In a mobile robot's case, the robot is operated in the actual environment it will operate in. The robot needs to perceive its environment and decide what its next course of action will be, and each action will be given a reward. The goal is to maximize the reward hence choose the best next state possible. Q-learning has been used to navigate robots in dynamic and, in some cases, unknown environments [322]. In order to improve efficiency, auxiliary tasks like value function [323] and prediction-based reward [324] has been added. Only visual-based solution [325] and visual plus language based solutions [326], [327] have also been researched. However, the shortcomings of RL have been its operation area being limited to small configuration spaces and fail to replicate a high level of complex performance in large action space.

The inclusion of Deep learning in the reward-based system has seen more promising results in robot navigation. Deep learning enables robots to perceive the environment at a higher level, learn the attributes of the sensor data automatically, which enables robots to navigate in more complex environments. Some wheel robot-based methods have been proposed which generalizes well across training and real-life scenarios and can deal with noisy data, and well versed in obstacle avoidance capabilities [328], [329]. Initially, value-based DRL solutions were proposed to advance the navigation purpose of mobile robots. One example of it is the use of a Deep Q-network [329] to navigate a mobile robot in a corridor using only an RGB-D sensor. But this only works for discrete environments. TO work the robot continuously in much bigger environments, policy-based methods have been introduced which work on the basis of formulating and finishing policies rather than working for a reward. DDPG [329], NAF [330], multi-threading DDPG [331] etc are some of the continuous control policy-based methods. the combination of these two approaches has been later adopted in A3C network [332]. It constitutes an actor and a critic network which uses multi-threading to avoid overfitting. Zhu *et al.* [292] used A3C network in a target-driven approach using only visual sensors and solved the problem of

generalization and data inefficiency. This was later adopted by [333] to navigate in indoor environments. Quan *et al.* [334] proposed a novel architecture for robot navigation combining DDQN with Recurrent Neural Network, which improves global pathfinding capacities around 5 *percent* than other existing algorithms. Some other DDQN based methods are [335], [336] etc. Another proposed to combine CNN with DRL to predict the steering action of the robot more accurately while traversing [337].

VII. DISCUSSION ON FUTURE TRENDS OF MOBILE ROBOTICS

In every department of mobile robotics, there have been significant developments. From building 2D maps to 3D dense maps with rich semantic information, navigation in static mapped environments to dynamic unknown environments, performing in a case study scenario to actually getting used daily in real-life cases, it is clear that mobile robotics has come a long way, and integrating deep-learning seems to be the way to go forward in the future. While the integration of deep learning into robotics has caused significant improvement in some sectors, it also brought a set of new challenges going forward.

A. ACHIEVING A COMPLETE END-TO-END LEARNING SOLUTION

While deep learning solutions have brought significant upgrades to the existing systems, there has also been researching to use only deep learning models to complete a task such as SLAM or navigation. However, unfortunately, end-to-end learning based models have yet to reach the performers of hybrid models. Deep learning models are data-driven models; hence their performance is hugely affected by the size, quality, and features of the training data. There is also a concern of these approaches being a black-box solution in some ways as tuning these models to achieve generalized performance across any environment may be tricky, as we often are not sure tweaking which parameter will deliver better results. Hence going forward, achieving end-to-end models for navigation, SLAM, or odometry estimation seems to be an area of development to be worked on in the next decade.

B. DEVELOPING MODELS FOR EDGE DEVICES

In the last decade, one of the most promising sectors was decreasing the size of our computers while increasing power. We now have these small computers that can fit into our pockets. These are called edge devices and are perfect for mobile robotics. Thus, developers around the world have dedicated time and effort in developing applications that can run on these edge devices as it is of great convenience to run applications on the go rather than transferring data on the server for analysis. Some existing devices currently on the market are raspberry pi [338], Nvidia Zetson [339] etc. However, deep learning models need high-level GPU to run properly, which these edge devices cannot provide. Hence, optimized versions of the existing deep learning models are being developed. There is also research ongoing to lower the

cost of edge devices and increase their computing capabilities. In the next decade or so, developing optimized models which can work in real time in low-cost edge devices seems to be a real challenge.

C. LONG TIME CONSISTENCY

Though mobile robots are being used in many real-life cases in the current world, these robots still lack the capability to run fully autonomously consistently for a long time. As time goes on, the current generation of robots is prone to making errors. Thus regular maintenance and human intervention are necessary even in places where robots are supposed to be fully autonomous. The goal is to achieve fully autonomous capabilities for the life-cycle of the robot and minimizing human contacts as possible, reducing robot operators. These sectors still need lots of works going forward.

Apart from these, there are also some common issues: mapping complex environments and handling dynamic objects, dealing with uncertainty in environments, dealing with unintended failures, and integrating new sensors such as thermal camera, event camera, loop closure, map re-usability, and working with multi-agent. These are general problems in mobile robotics and still open topics for research in the future.

VIII. CONCLUSION

The modernization of the world makes the nation more dependent on autonomous robots to perform their daily routine. Nowadays, the mobile robot is becoming popular in both sectors of home and industrial purpose. Thus, the accuracy and characteristics of the operations of the mobile robot must be efficient and real-time. Any misleading information of the mobile robot about their working place may lead to serious damage to both the human life and the working place. Thus, the modeling and control approaches of the mobile robot must be efficient to handle any uncertainty or system dynamics. This research article has investigated all the factors of the mobile robot such as locomotion, perception, navigation, and cognition in a comparative model. These factors have been analyzed in terms of mathematical modeling, integration issues, critical challenges, and control issues. Different control approach in various sectors has been reviewed in this article to make the mobile robot more accurate and controllable. The review on the integration of deep learning in the mobile robot and the future trends will help the researchers to take this sector to a new level.

REFERENCES

[1] *Eufy Vacuum*. Accessed: Aug. 19, 2020. [Online]. Available: http://www.eufyvacuum.com/

[2] *Penny, One Robot, Versatile Tray System*. Accessed: Aug. 18, 2020. [Online]. Available: https://www.bearrobotics.ai/products/

[3] S. I. A. P. Diddeniya, W. K. I. L. Wanniarachchi, P. R. S. De Silva, and N. C. Ganegoda, "Efficient office assistant robot system: Autonomous navigation and controlling based on ROS," *Int. J. Multidisciplinary Stud.*, vol. 6, no. 1, p. 64, Jun. 2019.

[4] *SANBOT ELF*. Accessed: Aug. 19, 2020. [Online]. Available: https://robot.omitech.it/en/sanbot-elf/

[5] R. W. Dobbs, W. R. Halgrimson, S. Talamini, H. T. Vigneswaran, J. O. Wilson, and S. Crivellaro, "Single-port robotic surgery: The next generation of minimally invasive urology," *World J. Urol.*, vol. 38, no. 4, pp. 897–905, Apr. 2020.

[6] C. L. Stewart, P. H. G. Ituarte, K. A. Melstrom, S. G. Warner, L. G. Melstrom, L. L. Lai, Y. Fong, and Y. Woo, "Robotic surgery trends in general surgical oncology from the national inpatient sample," *Surgical Endoscopy*, vol. 33, no. 8, pp. 2591–2601, Aug. 2019.

[7] R. Bischoff, "From research to products: The development of the KUKA light-weight robot," in *Proc. 40th Int. Symp. Robot.*, Barcelona, Spain, 2009.

[8] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. Cambridge, MA, USA: MIT Press, 2011.

[9] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419839596.

[10] S. R. Fahim, M. Rashiduzzaman, S. Dey, M. S. Hosen, S. K. Sarker, and S. K. Das, "Modeling and development of a five DoF vision based remote operated robotic arm," in *Proc. IEEE Int. Conf. Robot., Autom., Artif.-Intell. Internet–Things (RAAICON)*, Nov. 2019, pp. 56–59.

[11] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2016.

[12] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[13] F. Gul, W. Rahiman, and S. S. Nazli Alhady, "A comprehensive study for robot navigation techniques," *Cogent Eng.*, vol. 6, no. 1, Jan. 2019, Art. no. 1632046.

[14] S. Huang and G. Dissanayake, "Robot localization: An introduction," *Wiley Encyclopedia Elect. Electron. Eng.*, pp. 1–10, 1999.

[15] S. Thrun, "Robotic mapping: A survey," *Exploring Artif. Intell. New Millennium*, vol. 1, nos. 1–35, p. 1, 2002.

[16] *Executive Summary World Robotics 2018 Service Robots*, International Federation of Robotics, Frankfurt, Germany, 2018.

[17] S. Onofre, B. Caseiro, J. P. Pimentão, and P. Sousa, "Using fuzzy logic to improve BLE indoor positioning system," in *Proc. Doctoral Conf. Comput., Elect. Ind. Syst.* Cham, Switzerland: Springer, 2016, pp. 169–177.

[18] H.-Y. Zhang, W.-M. Lin, and A.-X. Chen, "Path planning for the mobile robot: A review," *Symmetry*, vol. 10, no. 10, p. 450, Oct. 2018.

[19] C. Chen, B. Wang, C. Xiaoxuan Lu, N. Trigoni, and A. Markham, "A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence," 2020, *arXiv:2006.12567*. [Online]. Available: http://arxiv.org/abs/2006.12567

[20] I. Karabegović, E. Karabegović, M. Mahmić, and E. Husak, "Implementation of industry 4.0 and industrial robots in the manufacturing processes," in *Proc. Int. Conf. 'New Technol., Develop. Appl.'* Cham, Switzerland: Springer, 2019, pp. 3–14.

[21] *Penny: One Robot, Versatile Tray System*. Accessed: Nov. 25, 2020. [Online]. Available: https://www.bearrobotics.ai/products/

[22] *Temi the Personal Robot*. Accessed: Nov. 25, 2020. [Online]. Available: https://www.robotemi.com/

[23] *Educational Robot Snow*. Accessed: Jun. 21, 2020. [Online]. Available: https://mirobotic.sg/education-robot-snow/

[24] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[25] S. Dubey, M. Prateek, and M. Saxena, "Robot locomotion-a review," *Int. J. Appl. Eng. Res.*, vol. 10, no. 3, pp. 7357–7369, 2015.

[26] J. A. Grimes and J. W. Hurst, "The design of ATRIAS 1.0 a unique monopod, hopping robot," in *Adaptive Mobile Robotics*. Singapore: World Scientific, 2012, pp. 548–554.

[27] A. M. M. Omer, R. Ghorbani, H.-o. Lim, and A. Takanishi, "Semi-passive dynamic walking for biped walking robot using controllable joint stiffness based on dynamic simulation," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2009, pp. 1600–1605.

[28] T. Nagasaki, S. Kajita, K. Yokoi, K. Kaneko, and K. Tanie, "Running pattern generation and its evaluation using a realistic humanoid model," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Sep. 2003, pp. 1336–1342.

[29] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, and K. Tanie, "Planning walking patterns for a biped robot," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 280–289, Jun. 2001.

[30] S. Martel, M. Sherwood, C. Helm, W. Garcia de Quevedo, T. Fofonoff, R. Dyer, J. Bevilacqua, J. Kaufman, O. Roushdy, and I. Hunter, "Three-legged wireless miniature robots for mass-scale operations at the sub-atomic scale," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4, May 2001, pp. 3423–3428.

[31] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *J. Field Robot.*, vol. 35, no. 8, pp. 1311–1326, Dec. 2018.

[32] K. Arikawa and S. Hirose, "Development of quadruped walking robot TITAN-VIII," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 1, Nov. 1996, pp. 208–214.

[33] M. Raibert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the rough-terrain quadruped robot," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 10822–10825, 2008.

[34] C. Queiroz, N. Gonçalves, and P. Menezes, "A study on static gaits for a four leg robot," in *Proc. Control-UK ACC Int. Conf. Control*, 1999, pp. 1–6.

[35] A. Bouman, M. Fadhil Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, and A.-A. Agha-mohammadi, "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," 2020, *arXiv:2010.09259*. [Online]. Available: http://arxiv.org/abs/2010.09259

[36] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *Int. J. Robot. Res.*, vol. 32, no. 8, pp. 932–950, Jul. 2013.

[37] Z. Xu, T. Lü, and F. Ling, "Trajectory planning of jumping over obstacles for hopping robot," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 30, no. 4, pp. 327–334, Dec. 2008.

[38] A. Allafi, "Control of multi-link one-legged hopping locomotion," Ph.D. dissertation, Dept. Mech. Eng., Michigan State Univ., East Lansing, MI, USA, 2020.

[39] K. Sakamoto, M. Otsuki, T. Maeda, K. Yoshikawa, and T. Kubota, "Evaluation of hopping robot performance with novel foot pad design on natural terrain for hopper development," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3294–3301, Oct. 2019.

[40] T. Li, R. Calandra, D. Pathak, Y. Tian, F. Meier, and A. Rai, "Planning in learned latent action spaces for generalizable legged locomotion," 2020, *arXiv:2008.11867*. [Online]. Available: http://arxiv.org/abs/2008.11867

[41] E. Massi, L. Vannucci, U. Albanese, M. C. Capolei, A. Vandesompele, G. Urbain, A. M. Sabatini, J. Dambre, C. Laschi, S. Tolu, and E. Falotico, "Combining evolutionary and adaptive control strategies for quadruped robotic locomotion," *Frontiers Neurorobot.*, vol. 13, p. 71, Aug. 2019.

[42] A. Mahapatra, S. S. Roy, and D. K. Pratihar, "Multi-legged robots—A review," in *Multi-body Dynamic Modeling of Multi-legged Robots*. Singapore: Springer, 2020, pp. 11–32.

[43] E. C. Orozco-Madaleno, F. Gomez-Bravo, E. Castillo, and G. Carbone, "Evaluation of locomotion performances for a mecanum-wheeled hybrid hexapod robot," *IEEE/ASME Trans. Mechatronics*, early access, Sep. 28, 2021.

[44] S. Ajisaka, S. Nakamura, T. Kubota, and H. Hashimoto, "A new design and development of portable personal vehicle," *J. Asian Electr. Vehicles*, vol. 12, no. 2, pp. 1693–1698, 2014.

[45] A. J. Álvares, G. F. Andriolli, P. R. Dutra, M. M. De Sousa, and J. C. Ferreira, "A navigation and path planning system for the Nomad XR4000 mobile robot with remote Web monitoring," in *Proc. ABCM Symp. Mechatronics*, vol. 1, 2004, pp. 18–24.

[46] G. Campion, G. Bastin, and B. Dandrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. Robot. Autom.*, vol. 12, no. 1, pp. 47–62, Feb. 1996.

[47] S. Ostrovskaya, "Dynamics of quasiholonomic and nonholonomic reconfigurable rolling robots," Ph.D. dissertation, Dept. Mech. Eng., McGill Univ., Montreal, QC, Canada, 2001.

[48] T. B. Lauwers, G. A. Kantor, and R. L. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 2884–2889.

[49] P. Fankhauser and C. Gwerder, "Modeling and control of a ballbot," M.S. thesis, Eidgenössische Technische Hochschule Zürich, Zürich, Switzerland, 2010.

[50] M. Mori and S. Hirose, "Development of active cord mechanism ACM-R3 with agile 3D mobility," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Expanding Societal Role Robot. Next Millennium*, vol. 3, Oct./Nov. 2001, pp. 1552–1557.

[51] A. H. Al-dabbagh and R. Ronsse, "A review of terrain detection systems for applications in locomotion assistance," *Robot. Autonomous Syst.*, vol. 133, Nov. 2020, Art. no. 103628.

[52] *HallucII: The Next-Generation Robotic Vehicle*. Accessed: Aug. 18, 2020. [Online]. Available: https://furo.org/en/works/halluc2.html

[53] T. Ran, L. Yuan, and J. B. Zhang, "Scene perception based visual navigation of mobile robot in indoor environment," *ISA Trans.*, vol. 109, pp. 389–400, Mar. 2021.

[54] A. Taylor and L. D. Riek, "Robot perception of human groups in the real world: State of the art," in *Proc. AAAI Fall Symposia*, 2016, pp. 1–7.

[55] N. Bore, P. Jensfelt, and J. Folkesson, "Multiple object detection, tracking and long-term dynamics learning in large 3D maps," 2018, *arXiv:1801.09292*. [Online]. Available: http://arxiv.org/abs/1801.09292

[56] C. Dondrup *et al.*, "Real-time multisensor people tracking for human-robot spatial interaction," in *Proc. Workshop Mach. Learn. Social Robot. (ICRA)*, Seattle, WA, USA, May 2015.

[57] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved RGBD-based object detection and pose estimation for warehouse Pick-and-Place," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 1179–1185, Jul. 2016.

[58] S. J. Russell and J. Stuart, "Norvig," *Artificial Intelligence: A Modern Approach*. 2003, pp. 111–114.

[59] B. Arad, J. Balendonck, R. Barth, O. Ben-Shahar, Y. Edan, T. Hellström, J. Hemming, P. Kurtser, O. Ringdahl, T. Tielen, and B. Tuijl, "Development of a sweet pepper harvesting robot," *J. Field Robot.*, vol. 37, no. 6, pp. 1027–1039, Sep. 2020.

[60] Z. Liu, W. Liu, X. Gong, and J. Wu, "Simplified attitude determination algorithm using accelerometer and magnetometer with extremely low execution time," *J. Sensors*, vol. 2018, pp. 1–11, Nov. 2018.

[61] S. Dong, W. Li, Z. An, and Z. Xie, "A new master control framework for the next generation service robots based on ROS," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 5164–5168.

[62] D. H. Gawali and V. M. Wadhai, "Recent trends in energy management of wireless wearable bio sensor design," in *Proc. 11th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2017, pp. 408–412.

[63] K. Yovchev, D. Chikurtev, N. Chivarov, and M. Grueva, "An intelligent control system for service robots," *IFAC-PapersOnLine*, vol. 52, no. 25, pp. 327–332, 2019.

[64] *Kinect*. Accessed: Aug. 18, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Kinect

[65] M. Kamruzzaman and M. Tareq, "Design and implementation of a robotic technique based waiter," in *Proc. 3rd Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Dec. 2017, pp. 1–5.

[66] J. Iqbal, R. Xu, S. Sun, and C. Li, "Simulation of an autonomous mobile robot for LiDAR-based in-field phenotyping and navigation," *Robotics*, vol. 9, no. 2, p. 46, Jun. 2020.

[67] S. Coşar, M. Fernandez-Carmona, R. Agrigoroaie, J. Pages, F. Ferland, F. Zhao, S. Yue, N. Bellotto, and A. Tapus, "ENRICHME: Perception and Interaction of an Assistive Robot for the Elderly at Home," *Int. J. Social Robot.*, vol. 12, pp. 779–805, Feb. 2020.

[68] S. Pendleton, T. Uthaicharoenpong, Z. J. Chong, G. Ming James Fu, B. Qin, W. Liu, X. Shen, Z. Weng, C. Kamin, M. A. Ang, L. T. Kuwae, K. A. Marczuk, H. Andersen, M. Feng, G. Butron, Z. Z. Chong, M. H. Ang, E. Frazzoli, and D. Rus, "Autonomous golf cars for public trial of mobility-on-demand service," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 1164–1171.

[69] M. Kam, X. Zhu, and P. Kalata, "Sensor fusion for mobile robot navigation," *Proc. IEEE*, vol. 85, no. 1, pp. 108–119, Jan. 1997.

[70] M. Turan, Y. Almalioglu, H. Gilbert, H. Araujo, T. Cemgil, and M. Sitti, "EndoSensorFusion: Particle filtering-based multi-sensory data fusion with switching state-space model for endoscopic capsule robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–8.

[71] *RPlidar A1, 360 Laser Range Scanner*. Accessed: Aug. 18, 2020. [Online]. Available: https://www.slamtec.com/en/Lidar/A1

[72] A. Sinha, H. Chen, D. G. Danu, T. Kirubarajan, and M. Farooq, "Estimation and decision fusion: A survey," *Neurocomputing*, vol. 71, nos. 13–15, pp. 2650–2656, Aug. 2008.

[73] H. Kawasaki, T. Komatsu, and K. Uchiyama, "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II," *IEEE/ASME Trans. Mechatronics*, vol. 7, no. 3, pp. 296–303, Sep. 2002.

[74] D. Ranasinghe, K. Vidanapathirana, T. Wickramarachchi, K. Katuwandeniya, P. Jayasekara, and S. Ajisaka, "Development of a lightweight, low-cost, self-balancing personal mobility vehicle for autonomous indoor navigation," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2019, pp. 571–576.

[75] A. S. Al-Araji, A. K. Ahmed, and K. E. Dagher, "A cognition path planning with a nonlinear controller design for wheeled mobile robot based on an intelligent algorithm," *J. Eng.*, vol. 25, no. 1, pp. 64–83, Dec. 2018.

[76] B. Yamauchi, A. Schultz, and W. Adams, "Integrating exploration and localization for mobile robots," *Adapt. Behav.*, vol. 7, no. 2, pp. 217–229, Mar. 1999.

[77] A. B. Camargo, Y. Liu, G. He, and Y. Zhuang, "Mobile robot autonomous exploration and navigation in large-scale indoor environments," in *Proc. 10th Int. Conf. Intell. Control Inf. Process. (ICICIP)*, Dec. 2019, pp. 106–111.

[78] Y. Tang, C. Zhao, J. Wang, C. Zhang, Q. Sun, W. Zheng, W. Du, F. Qian, and J. Kurths, "An overview of perception and decision-making in autonomous systems in the era of learning," 2020, *arXiv:2001.02319*. [Online]. Available: http://arxiv.org/abs/2001.02319

[79] L. Wang, L. Zhao, G. Huo, R. Li, Z. Hou, P. Luo, Z. Sun, K. Wang, and C. Yang, "Visual semantic navigation based on deep learning for indoor mobile robots," *Complexity*, vol. 2018, Apr. 2018, Art. no. 1627185.

[80] Y. Xiang and D. Fox, "DA-RNN: Semantic mapping with data associated recurrent neural networks," 2017, *arXiv:1703.03098*. [Online]. Available: http://arxiv.org/abs/1703.03098

[81] D. Filliat and J.-A. Meyer, "Map-based navigation in mobile robots: I. A review of localization strategies," *Cogn. Syst. Res.*, vol. 4, no. 4, pp. 243–282, 2003.

[82] T. Einsele, "Real-time self-localization in unknown indoor environment using a panorama laser range finder," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot Syst. Innov. Robot. Real-World Appl. (IROS)*, vol. 2, Sep. 1997, pp. 697–702.

[83] G. Dudek and P. MacKenzie, "Model-based map construction for robot localization," in *Proc. Vis. Interface*, no. 2, 1993, pp. 1–6.

[84] J. Jiang and J. Xin, "Path planning of a mobile robot in a free-space environment using Q-learning," *Prog. Artif. Intell.*, vol. 8, no. 1, pp. 133–142, Apr. 2019.

[85] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Proc. Workshop Dyn. Vis. (ICCV)*, vol. 20. Rio de Janeiro, Brazil, 2007, pp. 1–12.

[86] J. Zubizarreta, I. Aguinaga, and J. M. M. Montiel, "Direct sparse mapping," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1363–1370, Aug. 2020.

[87] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 834–849.

[88] A. Pronobis, "Semantic mapping with mobile robots," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, 2011.

[89] D. Lang and D. Paulus, "Semantic maps for robotics," in *Proc. Workshop Workshop AI Robot. (ICRA)*, Chicago, IL, USA, 2014, pp. 14–18.

[90] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[91] M. Colledanchise, D. Malafronte, and L. Natale, "Act, perceive, and plan in belief space for robot localization," 2020, *arXiv:2002.08124*. [Online]. Available: http://arxiv.org/abs/2002.08124

[92] Y. Chen, W. Chen, L. Zhu, Z. Su, X. Zhou, Y. Guan, and G. Liu, "A study of sensor-fusion mechanism for mobile robot global localization," *Robotica*, vol. 37, no. 11, pp. 1835–1849, Nov. 2019.

[93] S. Diamantas, "Towards resolving the kidnapped robot problem: Topological localization from crowdsourcing and georeferenced images," in *Proc. Sci. Inf. Conf.* Cham, Switzerland: Springer, 2019, pp. 551–563.

[94] C. Yi, "Map representation for robots," *Smart Comput. Rev.*, vol. 2, no. 1, pp. 18–27, Feb. 2012.

[95] R. Singh and K. S. Nagla, "Comparative analysis of range sensors for the robust autonomous navigation—A review," *Sensor Rev.*, 2019.

[96] O. Manu, "A study of indoor localization techniques," *DOCT-US*, vol. 1, no. 2, p. 32, 2009.

[97] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1327–1346, 2nd Quart., 2017.

[98] M. Maheepala, A. Z. Kouzani, and M. A. Joordens, "Light-based indoor positioning systems: A review," *IEEE Sensors J.*, vol. 20, no. 8, pp. 3971–3995, Apr. 2020.

[99] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *J. Robotic Syst.*, vol. 14, no. 4, pp. 231–249, Apr. 1997.

[100] X. Wang, X. Wang, and D. M. Wilkes, *Machine Learning-based Natural Scene Recognition for Mobile Robot Localization in An Unknown Environment*. Springer, 2019.

[101] X.-X. Chen and J. Huang, "Odor source localization algorithms on mobile robots: A review and future outlook," *Robot. Auto. Syst.*, vol. 112, pp. 123–136, Feb. 2019.

[102] J.-S. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Sep./Oct. 2002, pp. 454–459.

[103] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *Proc. 8th Int. Conf. Intell. Netw. Intell. Syst. (ICINIS)*, Nov. 2015, pp. 74–77.

[104] Y. Kim and H. Bang, "Introduction to Kalman filter and its applications," in *Introduction Implementations Kalman Filter*. Rijeka, Croatia: InTechOpen, 2018.

[105] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. and Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA, USA: MIT Press, 2005.

[106] P. G. Medewar, M. Yadav, and H. G. Patel, "A comparison between nonlinear estimation based algorithms for mobile robot localizations," in *Proc. IEEE 1st Int. Conf. Energy, Syst. Inf. Process. (ICESIP)*, Jul. 2019, pp. 1–6.

[107] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *J. Artif. Intell. Res.*, vol. 11, pp. 391–427, Nov. 1999.

[108] D. Fox, S. Thrun, and W. Burgard, "Active Markov localization for mobile robots," *Robot. Auton. Syst.*, vol. 25, nos. 3–4, pp. 195–207, 1998.

[109] F. R. Q. Aini, A. N. Jati, and U. Sunarya, "A study of Monte Carlo localization on robot operating system," in *Proc. Int. Conf. Inf. Technol. Syst. Innov. (ICITSI)*, Oct. 2016, pp. 1–6.

[110] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, May 1999, pp. 1322–1328.

[111] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: Efficient position estimation for mobile robots," *Proc. AAAI/IAAI*, nos. 343–349, 1999, p. 2.

[112] F. Boniardi, T. Caselitz, R. Kümmerle, and W. Burgard, "A pose graph-based localization system for long-term navigation in CAD floor plans," *Robot. Auto. Syst.*, vol. 112, pp. 84–97, Feb. 2019.

[113] X. Fang, C. Wang, T.-M. Nguyen, and L. Xie, "Graph optimization approach to range-based localization," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jan. 22, 2020, doi: 10.1109/TSMC.2020.2964713.

[114] G. Grisetti, C. Stachniss, and W. Burgard, "Nonlinear constraint network optimization for efficient map learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 428–439, Sep. 2009.

[115] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 2262–2269.

[116] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "ISAM2: Incremental smoothing and mapping using the bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, Feb. 2012.

[117] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G²o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 3607–3613.

[118] L. Carlone and F. Dellaert, "Duality-based verification techniques for 2D SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 4589–4596.

[119] L. Carlone, G. C. Calafiore, C. Tommolillo, and F. Dellaert, "Planar pose graph optimization: Duality, optimal solutions, and verification," *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 545–565, Jun. 2016.

[120] L. Carlone and G. C. Calafiore, "Convex relaxations for pose graph optimization with outliers," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1160–1167, Apr. 2018.

[121] V. Indelman, E. Nelson, N. Michael, and F. Dellaert, "Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 593–600.

[122] S. Kwon, K. Yang, and S. Park, "An effective Kalman filter localization method for mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 1524–1529.

[123] R. Negenborn, "Robot localization and Kalman filters," Utrecht Univ., Utrecht, The Netherlands, Tech. Rep. INF/SCR-0309, 2003.

[124] L. D'Alfonso, W. Lucia, P. Muraca, and P. Pugliese, "Mobile robot localization via EKF and UKF: A comparison based on real data," *Robot. Auto. Syst.*, vol. 74, pp. 122–127, Dec. 2015.

[125] F. Martinelli, "Robot localization: Comparable performance of EKF and UKF in some interesting indoor settings," in *Proc. 16th Medit. Conf. Control Autom.*, Jun. 2008, pp. 499–504.

[126] S. I. Roumeliotis and G. A. Bekey, "Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia (ICRA)*, vol. 3, Apr. 2000, pp. 2985–2992.

[127] J. Miura and K. Yamamoto, "Robust view matching-based Markov localization in outdoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 2970–2976.

[128] T. Naseer, B. Suger, M. Ruhnke, and W. Burgard, "Vision-based Markov localization for long-term autonomy," *Robot. Auto. Syst.*, vol. 89, pp. 147–157, Mar. 2017.

[129] J.-S. Gutmann, "Markov-Kalman localization for mobile robots," in *Proc. Object Recognit. Supported User Interact. Service Robots*, vol. 2, Aug. 2002, pp. 601–604.

[130] S. Xu, W. Chou, and H. Dong, "A robust indoor localization system integrating visual localization aided by CNN-based image retrieval with Monte Carlo localization," *Sensors*, vol. 19, no. 2, p. 249, Jan. 2019.

[131] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell.*, vol. 128, nos. 1–2, pp. 99–141, May 2001.

[132] P. Pfaff, W. Burgard, and D. Fox, "Robust Monte-Carlo localization using adaptive likelihood models," in *Proc. Eur. Robot. Symp.* Berlin, Germany: Springer, 2006, pp. 181–194.

[133] L. Zhang, R. Zapata, and P. Lepinay, "Self-adaptive Monte Carlo localization for mobile robots using range sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 1541–1546.

[134] C. Stachniss, *Robotic Mapping Exploration*, vol. 55. Springer, 2009.

[135] S. B. Samsuri, H. Zamzuri, M. A. A. Rahman, S. A. Mazlan, and A. H. A. Rahman, "Computational cost analysis of extended Kalman filter in simultaneous localization and mapping (EKF-SLAM) problem for autonomous vehicle," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 17, pp. 153–158, 2015.

[136] V. Imani, K. Haataja, and P. Toivanen, "Three main paradigms of simultaneous localization and mapping (SLAM) problem," *Proc. SPIE*, vol. 10696, Apr. 2018, Art. no. 106961P.

[137] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 1, Oct. 2003, pp. 206–211.

[138] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[139] C. Stachniss, G. Grisetti, W. Burgard, and N. Roy, "Analyzing Gaussian proposal distributions for mapping with rao-blackwellized particle filters," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2007, pp. 3485–3490.

[140] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi, "Fast and accurate SLAM with Rao–Blackwellized particle filters," *Robot. Auton. Syst.*, vol. 55, no. 1, pp. 30–38, 2007.

[141] G. Vallicrosa and P. Ridao, "H-SLAM: Rao-blackwellized particle filter SLAM using Hilbert maps," *Sensors*, vol. 18, no. 5, p. 1386, May 2018.

[142] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, winter 2010.

[143] M. Golfarelli, D. Maio, and S. Rizzi, "Elastic correction of dead-reckoning errors in map building," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Innov. Theory, Pract. Appl.*, vol. 2, Oct. 1998, pp. 905–911.

[144] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Inst. Technol., Atlanta, GA, USA, Tech. Rep., 2012.

[145] G. Grisetti, R. Kummerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 273–278.

[146] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time dynamic 3D surface reconstruction and interaction," in *ACM SIGGRAPH Talks*, 2011, p. 23.

[147] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

[148] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," in *Robotics: Science and Systems*, vol. 3, W. Burgard, O. Brock, and C. Stachniss, Eds. 2007, p. 9.

[149] T. Fan, H. Wang, M. Rubenstein, and T. Murphey, "CPL-SLAM: Efficient and certifiably correct planar graph-based SLAM using the complex number representation," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1719–1737, Dec. 2020.

[150] K. Koide, J. Miura, M. Yokozuka, S. Oishi, and A. Banno, "Interactive 3D graph SLAM for map correction," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 40–47, Jan. 2021.

[151] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 242–257, Jun. 2001.

[152] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel, "1-point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry," *J. Field Robot.*, vol. 27, no. 5, pp. 609–631, Sep. 2010.

[153] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2432–2437.

[154] M. Montemerlo and S. Thrun, "FastSLAM 1.0," in *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. 2007, pp. 27–62.

[155] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. IJCAI*, 2003, pp. 1151–1156.

[156] A. Eliazar and R. Parr, "DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks," in *Proc. IJCAI*, vol. 3. Acapulco, Mexico, 2003, pp. 1135–1142.

[157] G. Dubbelman and B. Browning, "COP-SLAM: Closed-form online pose-chain optimization for visual SLAM," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1194–1213, Oct. 2015.

[158] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1271–1278.

[159] J. Li, H. Zhan, B. M. Chen, I. Reid, and G. H. Lee, "Deep learning for 2D scan matching and loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 763–768.

[160] K. Lenac, A. Kitanov, R. Cupec, and I. Petrović, "Fast planar surface 3D SLAM using LIDAR," *Robot. Auto. Syst.*, vol. 92, pp. 197–220, Jun. 2017.

[161] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga, "Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation," *J. Field Robot.*, vol. 27, no. 1, pp. 52–84, Jan. 2010.

[162] J. Xiao, J. Zhang, B. Adler, H. Zhang, and J. Zhang, "Three-dimensional point cloud plane segmentation in both structured and unstructured environments," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1641–1652, Dec. 2013.

[163] M. Vlaminck, H. Luong, and W. Philips, "Have i seen this place before? A fast and robust loop detection and correction method for 3D lidar SLAM," *Sensors*, vol. 19, no. 1, p. 23, Dec. 2018.

[164] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, and F. Fraunhofer, "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proc. Robot., Sci. Syst. (RSS)*, 2020.

[165] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 55–81, Jan. 2015.

[166] X. Gao, *The Fourteen Lectures on Visual SLAM*. Beijing, China: Publishing House Electron. Ind., 2017, pp. 16–18.

[167] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[168] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2006, pp. 404–417.

[169] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[170] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[171] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[172] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2320–2327.

[173] R. A. Newcombe, A. Fitzgibbon, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, and S. Hodges, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 127–136.

[174] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *Proc. Robot., Sci. Syst. (RSS) Workshop RGB-D, Adv. Reasoning Depth Cameras*, Jul. 2012.

[175] T. Schops, T. Sattler, and M. Pollefeys, "BAD SLAM: Bundle adjusted direct RGB-D SLAM," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 134–144.

[176] Y. Chen, Y. Zhou, Q. Lv, and K. K. Deveerasetty, "A review of V-SLAM*," in *Proc. IEEE Int. Conf. Inf. Automat. (ICIA)*, Aug. 2018, pp. 603–608.

[177] J. K. Makhubela, T. Zuva, and O. Y. Agunbiade, "A review on vision simultaneous localization and mapping (VSLAM)," in *Proc. Int. Conf. Intell. Innov. Comput. Appl. (ICONIC)*, Dec. 2018, pp. 1–5.

[178] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: A survey from 2010 to 2016," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 1, p. 16, Dec. 2017.

[179] A. Li, X. Ruan, J. Huang, X. Zhu, and F. Wang, "Review of vision-based simultaneous localization and mapping," in *Proc. IEEE 3rd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Mar. 2019, pp. 117–123.

[180] A. R. Soltani, H. Tawfik, J. Y. Goulermas, and T. Fernando, "Path planning in construction sites: Performance evaluation of the Dijkstra, A*, and GA search algorithms," *Adv. Eng. Inform.*, vol. 16, no. 4, pp. 291–303, 2002.

[181] H. Shi, W. Cao, S.-L. Zhu, and B.-S. Zhu, "Application of an improved A* algorithm in shortest route planning," *Geomatics Spatial Inf. Technol.*, vol. 32, no. 6, pp. 208–211, 2009.

[182] E. Masehian and M. R. Amin-Naseri, "A Voronoi diagram-visibility graph-potential field compound algorithm for robot path planning," *J. Robotic Syst.*, vol. 21, no. 6, pp. 275–300, 2004.

[183] E. J. Gomez, F. M. Santa, and F. H. M. Sarmiento, "A comparative study of geometric path planning methods for a mobile robot: Potential field and Voronoi diagrams," in *Proc. II Int. Congr. Eng. Mechatronics Autom. (CIIMA)*, Oct. 2013, pp. 1–6.

[184] P. Li, X. Huang, and M. Wang, "A novel hybrid method for mobile robot path planning in unknown dynamic environment based on hybrid DSm model grid map," *J. Exp. Theor. Artif. Intell.*, vol. 23, no. 1, pp. 5–22, Mar. 2011.

[185] I. Engedy and G. Horvath, "Artificial neural network based local motion planning of a wheeled mobile robot," in *Proc. 11th Int. Symp. Comput. Intell. Informat. (CINTI)*, Nov. 2010, pp. 213–218.

[186] M. B. Montaner and A. Ramirez-Serrano, "Fuzzy knowledge-based controller design for autonomous robot navigation," *Expert Syst. Appl.*, vol. 14, nos. 1–2, pp. 179–186, Jan. 1998.

[187] A. Zhu and S. X. Yang, "An adaptive neuro-fuzzy controller for robot navigation," in *Recent Advances in Intelligent Control Systems*. London, U.K.: Springer, 2009, pp. 277–307.

[188] S. Ahmadzadeh and M. Ghanavati, "Navigation of mobile robot using the PSO particle swarm optimization," *J. Academic Appl. Stud. (JAAS)*, vol. 2, no. 1, pp. 32–38, 2012.

[189] A. Ghorbani, S. Shiry, and A. Nodehi, "Using genetic algorithm for a mobile robot path planning," in *Proc. Int. Conf. Future Comput. Commun.*, Apr. 2009, pp. 164–166.

[190] M. A. P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 1102–1110, Jun. 2009.

[191] L. Palmieri and K. O. Arras, "A novel RRT extend function for efficient and smooth mobile robot motion planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 205–211.

[192] X. Wang, G. Zhang, J. Zhao, H. Rong, F. Ipate, and R. Lefticaru, "A modified membrane-inspired algorithm based on particle swarm optimization for mobile robot path planning," *Int. J. Comput. Commun. Control*, vol. 10, no. 5, pp. 732–745, 2015.

[193] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "MOSFLA-MRPP: Multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning," *Eng. Appl. Artif. Intell.*, vol. 44, pp. 123–136, Sep. 2015.

[194] A. Atyabi and D. M. Powers, "Review of classical and heuristic-based navigation and path planning approaches," *Int. J. Advancements Comput. Technol.*, vol. 5, no. 14, p. 1, 2013.

[195] J.-C. Latombe, *Robot Motion Planning*, vol. 124. Springer, 2012.

[196] Q. Zhang, D. Chen, and T. Chen, "An obstacle avoidance method of soccer robot based on evolutionary artificial potential field," *Energy Procedia*, vol. 16, pp. 1792–1798, 2012.

[197] R. Bohlin and L. E. Kavraki, "Path planning using lazy PRM," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia (ICRA)*, vol. 1, Apr. 2000, pp. 521–528.

[198] B. Ichter, E. Schmerling, T.-W.-E. Lee, and A. Faust, "Learned critical probabilistic roadmaps for robotic motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9535–9541.

[199] P. Bhattacharya and M. L. Gavrilova, "Roadmap-based path planning-using the Voronoi diagram for a clearance-based shortest path," *IEEE Robot. Automat. Mag.*, vol. 15, no. 2, pp. 58–66, Jun. 2008.

[200] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968.

[201] A. Stentz, "Optimal and efficient path planning for partially known environments," in *Intelligent Unmanned Ground Vehicles*. Boston, MA, USA: Springer, 1997, pp. 203–220.

[202] S. Koenig and M. Likhachev, "D* lite," in *Proc. AAAI/IAAI*, vol. 15, 2002.

[203] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*. Cham, Switzerland: Springer, 2019, pp. 43–55.

[204] H. Guo, Z. Mao, W. Ding, and P. Liu, "Optimal search path planning for unmanned surface vehicle based on an improved genetic algorithm," *Comput. Electr. Eng.*, vol. 79, Oct. 2019, Art. no. 106467.

[205] T. Guan-Zheng, H. Huan, and A. Sloman, "Ant colony system algorithm for real-time globally optimal path planning of mobile robots," *Acta Automatica Sinica*, vol. 33, no. 3, pp. 279–285, 2007.

[206] S. Liu, L. Mao, and J. Yu, "Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems," in *Proc. Int. Conf. Mechatronics Autom.*, Jun. 2006, pp. 1733–1738.

[207] J. Liu, J. Yang, H. Liu, X. Tian, and M. Gao, "An improved ant colony algorithm for robot path planning," *Soft Comput.*, vol. 21, no. 19, pp. 5829–5839, Oct. 2017.

[208] Y. Peng, S.-W. Li, and Z.-Z. Hu, "A self-learning dynamic path planning method for evacuation in large public buildings based on neural networks," *Neurocomputing*, vol. 365, pp. 71–85, Nov. 2019.

[209] B. K. Patle, G. Babu L, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.*, vol. 15, no. 4, pp. 582–606, Aug. 2019.

[210] A. W. Mohemmed, N. C. Sahoo, and T. K. Geok, "Solving shortest path problem using particle swarm optimization," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1643–1653, Sep. 2008.

[211] Y. Zhang, D.-W. Gong, and J.-H. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, Mar. 2013.

[212] I. Pérez-Hurtado, M. Á. Martínez-del-Amor, G. Zhang, F. Neri, and M. J. Pérez-Jiménez, "A membrane parallel rapidly-exploring random tree algorithm for robotic motion planning," *Integr. Comput.-Aided Eng.*, pp. 1–18, 2020.

[213] A. Al-Kaff, J. M. Armingol, and A. de La Escalera, "A vision-based navigation system for unmanned aerial vehicles (UAVs)," *Integr. Comput.-Aided Eng.*, vol. 26, no. 3, pp. 297–310, Apr. 2019.

[214] E. A. Bhuiyan, M. M. A. Akhand, S. K. Das, M. F. Ali, Z. Tasneem, M. R. Islam, D. Saha, F. R. Badal, M. H. Ahamed, and S. Moyeen, "A survey on fault diagnosis and fault tolerant methodologies for permanent magnet synchronous machines," *Int. J. Automat. Comput.*, vol. 17, pp. 763–787, Nov. 2020.

[215] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[216] N. Sleumer and N. Tschichold-Gürmann, "Exact cell decomposition of arrangements used for path planning in robotics," Dept. Comput. Sci., ETH Zürich, Zürich, Switzerland, Tech. Rep. 329, 1999.

[217] M. Barbehenn and S. Hutchinson, "Efficient search and hierarchical motion planning by dynamically maintaining single-source shortest paths trees," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 198–214, Apr. 1995.

[218] F. Lingelbach, "Path planning using probabilistic cell decomposition," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 1, Apr./May 2004, pp. 467–472.

[219] J.-W. Jung, B.-C. So, J.-G. Kang, D.-W. Lim, and Y. Son, "Expanded douglas–peucker polygonal approximation and opposite angle-based exact cell decomposition for path planning with curvilinear obstacles," *Appl. Sci.*, vol. 9, no. 4, p. 638, 2019.

[220] L. Zhou and W. Li, "Adaptive artificial potential field approach for obstacle avoidance path planning," in *Proc. 7th Int. Symp. Comput. Intell. Design*, vol. 2, Dec. 2014, pp. 429–432.

[221] U. Orozco-Rosas, O. Montiel, and R. Sepulveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Appl. Soft Comput.*, vol. 77, pp. 236–251, Apr. 2019.

[222] Y. Huang, H. Ding, Y. Zhang, H. Wang, D. Cao, N. Xu, and C. Hu, "A motion planning and tracking framework for autonomous vehicles based on artificial potential field elaborated resistance network approach," *IEEE Trans. Ind. Electron.*, vol. 67, no. 2, pp. 1376–1386, Feb. 2020.

[223] L. Lulu and A. Elnagar, "A comparative study between visibility-based roadmap path planning algorithms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Aug. 2005, pp. 3263–3268.

[224] J. A. Janet, R. C. Luo, and M. G. Kay, "The essential visibility graph: An approach to global motion planning for autonomous mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, May 1995, pp. 1958–1963.

[225] Y. Ma, G. Zheng, and W. Perruquetti, "Cooperative path planning for mobile robots based on visibility graph," in *Proc. 32nd Chin. Control Conf.*, Jul. 2013, pp. 4915–4920.

[226] C. Alexopoulos, "Path planning for a mobile robot," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 2, pp. 318–322, Mar. 1992.

[227] K. Jiang, L. D. Seneviratne, and S. Earles, "A shortest path based path planning algorithm for nonholonomic mobile robots," *J. Intell. Robotic Syst.*, vol. 24, no. 4, pp. 347–366, 1999.

[228] X. Yu, Y. Zhao, C. Wang, and M. Tomizuka, "Trajectory planning for robot manipulators considering kinematic constraints using probabilistic roadmap approach," *J. Dyn. Syst., Meas., Control*, vol. 139, no. 2, Feb. 2017, Art. no. 021001.

[229] F. A. Raheem and M. I. Abdulkareem, "Development of path planning algorithm using probabilistic roadmap based on modified ant colony optimization," *World J. Eng. Technol.*, vol. 7, no. 4, pp. 583–597, 2019.

[230] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[231] M. Przybylski and B. Putz, "D* extra lite: A dynamic A* with search–tree cutting and frontier–gap repairing," *Int. J. Appl. Math. Comput. Sci.*, vol. 27, no. 2, pp. 273–290, 2017.

[232] M. Pala, N. Eraghi, F. López-Colino, A. Sanchez, A. de Castro, and J. Garrido, "HCTNav: A path planning algorithm for low-cost autonomous robot navigation in indoor environments," *ISPRS Int. J. Geo-Inf.*, vol. 2, no. 3, pp. 729–748, Aug. 2013.

[233] N. O. Eraghi, F. López-Colino, A. de Castro, and J. Garrido, "NafisNav: An indoor navigation algorithm for embedded systems and based on grid maps," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*. Piscataway, NJ, USA: IEEE Press, 2015, pp. 345–350.

[234] J. Holland, "Adaptation in natural and artificial systems: An introductory analysis with application to biology," *Control Artif. Intell.*, to be published.

[235] G. Nagib and W. Gharieb, "Path planning for a mobile robot using genetic algorithms," in *Proc. IEEE Robot.*, Sep. 2004, pp. 185–189.

[236] C.-C. Tsai, H.-C. Huang, and C.-K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4813–4821, Oct. 2011.

[237] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Comput. Electr. Eng.*, vol. 38, no. 6, pp. 1564–1572, Nov. 2012.

[238] F. Liu, S. Liang, and D. X. Xian, "Optimal path planning for mobile robot using tailored genetic algorithm," *TELKOMNIKA Indonesian J. Electr. Eng.*, vol. 12, no. 1, pp. 1–9, Jan. 2014.

[239] S. Brahimi, O. Azouaoui, and M. Loudini, "Intelligent mobile robot navigation using a neuro-fuzzy approach," *Int. J. Comput. Aided Eng. Technol.*, vol. 11, no. 6, pp. 710–726, 2019.

[240] H. A. Ibrahim, A. T. Azar, Z. F. Ibrahim, and H. H. Ammar, "A hybrid deep learning based autonomous vehicle navigation and obstacles avoidance," in *Proc. Joint Eur.-US Workshop Appl. Invariance Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 296–307.

[241] H. K. Paikray, P. K. Das, S. Panda, and B. K. Balabantaray, "Improved shuffled frog leaping algorithm for path planning of multiple mobile-robot," in *Proc. 2nd Int. Conf. Innov. Electron., Signal Process. Commun. (IESC)*, Mar. 2019, pp. 132–137.

[242] J. Xin, S. Li, J. Sheng, Y. Zhang, and Y. Cui, "Application of improved particle swarm optimization for navigation of unmanned surface vehicles," *Sensors*, vol. 19, no. 14, p. 3096, Jul. 2019.

[243] F. Yan, "Distributed vision navigation technology for mobile robots based on jitter privacy ant colony algorithm," *Int. J. Elect. Eng. Educ.*, 2019, Art. no. 0020720918813810.

[244] N. Pérez-Higueras, A. Jardón, Á. Rodríguez, and C. Balaguer, "3D exploration and navigation with optimal-RRT planners for ground robots in indoor incidents," *Sensors*, vol. 20, no. 1, p. 220, Dec. 2019.

[245] B. B. K. Ayawli, X. Mei, M. Shen, A. Y. Appiah, and F. Kyeremeh, "Optimized RRT–A* path planning method for mobile robots in partially known environment," *Inf. Technol. Control*, vol. 48, no. 2, pp. 179–194, Jun. 2019.

[246] J. Qi, H. Yang, and H. Sun, "MOD-RRT*: A sampling-based algorithm for robot path planning in dynamic environment," *IEEE Trans. Ind. Electron.*, early access, Jun. 11, 2020, doi: 10.1109/TIE.2020.2998740.

[247] N. Chaiyaratana and A. M. S. Zalzala, "Time-optimal path planning and control using neural networks and a genetic algorithm," *Int. J. Comput. Intell. Appl.*, vol. 2, no. 2, pp. 153–172, Jun. 2002.

[248] P. Payeur, H. Le-Huy, and C. Gosselin, "Robot path planning using neural networks and fuzzy logic," in *Proc. 20th Annu. Conf. IEEE Ind. Electron. (IECON)*, vol. 2, Sep. 1994, pp. 800–805.

[249] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci. (MHS)*, 1995, pp. 39–43.

[250] V. Vonásek, J. Faigl, T. Krajník, and L. Přeučil, "RRT-path—a guided rapidly exploring random tree," in *Robot Motion and Control*, K. R. Kozłowski, Ed. London, U.K.: Springer, 2009, pp. 307–316.

[251] C. E. Tuncali and G. Fainekos, "Rapidly-exploring random trees for testing automated vehicles," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 661–666.

[252] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. Springer, 1986, pp. 396–404.

[253] R. C. Arkin, "Motor schema—Based mobile robot navigation," *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 92–112, 1989.

[254] D. Q. Bao and I. Zelinka, "Obstacle avoidance for swarm robot based on self-organizing migrating algorithm," in *Proc. Comput. Sci.*, vol. 150. Cham, Switzerland: Springer, 2019, pp. 425–432.

[255] M. Zohaib, S. M. Pasha, N. Javaid, and J. Iqbal, "IBA: Intelligent Bug Algorithm–A novel strategy to navigate mobile robots autonomously," in *Proc. Int. Multi Topic Conf.* Cham, Switzerland: Springer, 2013, pp. 291–299.

[256] S. Huang, R. S. H. Teo, and K. K. Tan, "Collision avoidance of multi unmanned aerial vehicles: A review," *Annu. Rev. Control*, vol. 48, pp. 147–164, 2019.

[257] S. Jin, M. U. Ahmed, J. W. Kim, Y. H. Kim, and P. K. Rhee, "Combining obstacle avoidance and visual simultaneous localization and mapping for indoor navigation," *Symmetry*, vol. 12, no. 1, p. 119, Jan. 2020.

[258] V. J. Lumelsky and T. Skewis, "Incorporating range sensing in the robot navigation function," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 5, pp. 1058–1069, Sep./Oct. 1990.

[259] V. J. Lumelsky and A. A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, vol. 2, nos. 1–4, pp. 403–430, 1987.

[260] M. Zohaib, S. Mustafa Pasha, N. Javaid, and J. Iqbal, "Intelligent bug algorithm (IBA): A novel strategy to navigate mobile robots autonomously," 2013, *arXiv:1312.4552*. [Online]. Available: http://arxiv.org/abs/1312.4552

[261] I. Kamon, E. Rivlin, and E. Rimon, "A new range-sensor based globally convergent navigation algorithm for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1, Apr. 1996, pp. 429–435.

[262] S. Y. Ng and N. S. Ahmad, "A bug-inspired algorithm for obstacle avoidance of a nonholonomic wheeled mobile robot with constraints," in *Proc. Intell. Comput.-Proc. Comput. Conf.* Cham, Switzerland: Springer, 2019, pp. 1235–1246.

[263] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[264] J. Borenstein and Y. Koren, "High-speed obstacle avoidance for mobile robots," in *Proc. IEEE Int. Symp. Intell. Control*, Aug. 1988, pp. 382–384.

[265] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1179–1187, Sep. 1989.

[266] I. Susnea, V. Minzu, and G. Vasiliu, "Simple, real-time obstacle avoidance algorithm for mobile robots," in *Proc. 8th WSEAS Int. Conf. Comput. Intell., Man-Mach. Syst. Cybern. (CIMMACS)*, 2009, pp. 24–29.

[267] N. Metsänen, "Assessment of local path planners in a indoor and outdoor robot," Tech. Rep., 2020.

[268] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, Apr. 1996, pp. 3375–3382.

[269] J. Lopez, C. Otero, R. Sanz, E. Paz, E. Molinos, and R. Barea, "A new approach to local navigation for autonomous driving vehicles based on the curvature velocity method," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 1751–1757.

[270] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[271] M. Zohaib, M. Pasha, R. A. Riaz, N. Javaid, M. Ilahi, and R. D. Khan, "Control strategies for mobile robot with obstacle avoidance," 2013, *arXiv:1306.1144*. [Online]. Available: http://arxiv.org/abs/1306.1144

[272] P. G. Luan and N. T. Thinh, "Real-time hybrid navigation system-based path planning and obstacle avoidance for mobile robots," *Appl. Sci.*, vol. 10, no. 10, p. 3355, 2020.

[273] H.-C. Chen, "Monocular vision-based obstacle detection and avoidance for a multicopter," *IEEE Access*, vol. 7, pp. 167869–167883, 2019.

[274] C. Schlegel, "Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Innov. Theory, Pract. Appl.*, vol. 1, Oct. 1998, pp. 594–599.

[275] J. Minguez and L. Montano, "Nearness diagram navigation (ND): A new real time collision avoidance approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Oct./Nov. 2000, pp. 2094–2100.

[276] C. S. Tzafestas and S. G. Tzafestas, "Recent algorithms for fuzzy and neurofuzzy path planning and navigation of autonomous mobile robots," *Syst. Sci.*, vol. 25, no. 2, pp. 25–39, 1999.

[277] C.-T. Chen, R. D. Quinn, and R. E. Ritzmann, "A crash avoidance system based upon the cockroach escape response circuit," in *Proc. Int. Conf. Robot. Autom.*, vol. 3, Apr. 1997, pp. 2007–2012.

[278] M. Demir and V. Sezer, "Autonomous overtaking maneuver design based on follow the gap method," in *Proc. 16th Int. Conf. Informat. Control, Autom. Robot.*, 2019, pp. 295–302.

[279] J. Mínguez and L. Montano, "Robot navigation in very complex, dense, and cluttered indoor/outdoor environments," *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 397–402, 2002.

[280] K. Konolige, "A gradient method for realtime robot control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 1, Oct./Nov. 2000, pp. 639–646.

[281] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, May 1998, pp. 1572–1577.

[282] I. Ulrich and J. Borenstein, "VFH*: Local obstacle avoidance with look-ahead verification," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia*, vol. 3, Apr. 2000, pp. 2505–2511.

[283] S. M. H. Rostami, A. K. Sangaiah, J. Wang, and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 70, Dec. 2019.

[284] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm: 'Follow the gap method,'" *Robot. Auton. Syst.*, vol. 60, no. 9, pp. 1123–1134, 2012.

[285] S.-A. Li, L.-H. Chou, T.-H. Chang, C.-H. Yang, and Y.-C. Chang, "Obstacle avoidance of mobile robot based on hyperomni vision," *Sensors Mater.*, vol. 31, no. 3, pp. 1021–1036, 2019.

[286] R. Singh and T. Bera, "Obstacle avoidance of mobile robot using fuzzy logic and hybrid obstacle avoidance algorithm," in *Proc. IOP Conf., Mater. Sci. Eng.*, vol. 517, no. 1. Bristol, U.K.: IOP Publishing, 2019, Art. no. 012009.

[287] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke, "The limits and potentials of deep learning for robotics," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 405–420, Apr. 2018.

[288] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4628–4635.

[289] L. Ma, J. Stuckler, C. Kerl, and D. Cremers, "Multi-view deep learning for consistent semantic mapping with RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 598–605.

[290] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2043–2050.

[291] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3 VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1281–1292.

[292] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3357–3364.

[293] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[294] P. Mirowski, M. K. Grimes, and M. Malinowski, "Learning to navigate in cities without a map," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2419–2430.

[295] K. R. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in *Proc. VISAPP*, vol. 1, 2015, pp. 486–490.

[296] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with CNNs for frame-to-frame ego-motion estimation," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 18–25, Jan. 2016.

[297] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 963–968.

[298] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[299] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.

[300] M. R. U. Saputra, P. P. B. de Gusmao, S. Wang, A. Markham, and N. Trigoni, "Learning monocular visual odometry through geometry-aware curriculum learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3549–3555.

[301] M. R. U. Saputra, P. Gusmao, Y. Almalioglu, A. Markham, and N. Trigoni, "Distilling knowledge from a deep pose regressor network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 263–272.

[302] F. Xue, X. Wang, S. Li, Q. Wang, J. Wang, and H. Zha, "Beyond tracking: Selecting memory and refining poses for deep visual odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8575–8583.

[303] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7286–7291.

[304] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1983–1992.

[305] 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. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 340–349.

[306] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-inertial odometry as a Sequence-to-Sequence learning problem," 2017, *arXiv:1701.08376*. [Online]. Available: http://arxiv.org/abs/1701.08376

[307] E. J. Shamwell, K. Lindgren, S. Leung, and W. D. Nothwang, "Unsupervised deep visual-inertial odometry with online error correction for RGB-D imagery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2478–2493, Oct. 2020.

[308] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni, "Selective sensor fusion for neural visual-inertial odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10542–10551.

[309] L. Han, Y. Lin, G. Du, and S. Lian, "DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3D geometric constraints," 2019, *arXiv:1906.11435*. [Online]. Available: http://arxiv.org/abs/1906.11435

[310] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2510–2517.

[311] R. Clark, M. Bloesch, J. Czarnowski, S. Leutenegger, and A. J. Davison, "LS-Net: Learning to solve nonlinear least squares for monocular stereo," 2018, *arXiv:1809.02966*. [Online]. Available: http://arxiv.org/abs/1809.02966

[312] C. Tang and P. Tan, "BA-net: Dense bundle adjustment network," 2018, *arXiv:1806.04807*. [Online]. Available: http://arxiv.org/abs/1806.04807

[313] J. J. Moré, "The Levenberg-Marquardt algorithm: Implementation and theory," in *Numerical Analysis*. Berlin, Germany: Springer, 1978, pp. 105–116.

[314] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6243–6252.

[315] H. Zhou, B. Ummenhofer, and T. Brox, "DeepTAM: Deep tracking and mapping," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 822–838.

[316] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, nos. 1–4, pp. 43–52, Dec. 2010.

[317] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Robotics: Science and Systems XI*. 2015, pp. 1–10.

[318] Q. Liu and F. Duan, "Loop closure detection using CNN words," *Intell. Service Robot.*, vol. 12, no. 4, pp. 303–318, Oct. 2019.

[319] X. Gao and T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," *Auto. Robots*, vol. 41, no. 1, pp. 1–18, Jan. 2017.

[320] A. R. Memon, H. Wang, and A. Hussain, "Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems," *Robot. Auto. Syst.*, vol. 126, Apr. 2020, Art. no. 103470.

[321] N. Merrill and G. Huang, "Lightweight unsupervised deep loop closure," 2018, *arXiv:1805.07703*. [Online]. Available: http://arxiv.org/abs/1805.07703

[322] M. A. Kareem Jaradat, M. Al-Rousan, and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 135–149, Feb. 2011.

[323] M. Bellemare, W. Dabney, R. Dadashi, A. A. Taiga, P. S. Castro, N. Le Roux, D. Schuurmans, T. Lattimore, and C. Lyle, "A geometric perspective on optimal representations for reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4358–4369.

[324] M. Jaderberg, V. Mnih, W. Marian Czarnecki, T. Schaul, J. Z Leibo, D. Silver, and K. Kavukcuoglu, "Reinforcement learning with unsupervised auxiliary tasks," 2016, *arXiv:1611.05397*. [Online]. Available: http://arxiv.org/abs/1611.05397

[325] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, and R. Hadsell, "Learning to navigate in complex environments," 2016, *arXiv:1611.03673*. [Online]. Available: http://arxiv.org/abs/1611.03673

[326] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6629–6638.

[327] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sunderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-Language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3674–3683.

[328] X. Chen, A. Ghadirzadeh, J. Folkesson, M. Bjorkman, and P. Jensfelt, "Deep reinforcement learning to acquire navigation skills for wheel-legged robots in complex environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3110–3116.

[329] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," 2017, *arXiv:1706.09829*. [Online]. Available: http://arxiv.org/abs/1706.09829

[330] S. Gu, T. Lillicrap, I. Sutskever, and S. Levine, "Continuous deep q-learning with model-based acceleration," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2829–2838.

[331] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 31–36.

[332] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[333] H. Surmann, C. Jestel, R. Marchel, F. Musberg, H. Elhadj, and M. Ardani, "Deep reinforcement learning for real autonomous mobile robot navigation in indoor environments," *arXiv preprint arXiv:2005.13857*, 2020.

[334] H. Quan, Y. Li, and Y. Zhang, "A novel mobile robot navigation method based on deep reinforcement learning," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 3, 2020, Art. no. 1729881420921672.

[335] Y. Wang, Y. Fang, P. Lou, J. Yan, and N. Liu, "Deep reinforcement learning based path planning for mobile robot in unknown environment," in *Proc. J. Phys., Conf.*, vol. 1576, no. 1. Bristol, U.K.: IOP Publishing, 2020, Art. no. 012009.

[336] K. Yokoyama and K. Morioka, "Autonomous mobile robot with simple navigation system based on deep reinforcement learning and a monocular camera," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Jan. 2020, pp. 525–530.

[337] G. Chen, L. Pan, Y. Chen, P. Xu, Z. Wang, P. Wu, J. Ji, and X. Chen, "Robot navigation with map-based deep reinforcement learning," 2020, *arXiv:2002.04349*. [Online]. Available: http://arxiv.org/abs/2002.04349

[338] *Raspberry Pi 4, Your Tiny, Dual-Display, Desktop Computer*. Accessed: Aug. 18, 2020. [Online]. Available: https://www.raspberrypi.org/

[339] *JETSON XAVIER NX, The World's Smallest AI Supercomputer for Embedded and Edge Systems*, (accessed: 08.18.2020). Accessed: Aug. 18, 2020. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/

MD. A. K. NILOY is currently pursuing the B.Sc. degree in mechatronics engineering with the Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bangladesh. His research interests include autonomous mobile robots, SLAM, mechatronics systems, machine learning, and deep learning.

ANIKA SHAMA is currently pursuing the B.Sc. degree in mechatronics engineering with the Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bangladesh. Her research interests include autonomous mobile robots, mechatronics systems, electric vehicles control, and computer aided design.

RIPON K. CHAKRABORTTY (Member, IEEE) received the B.Sc. and M.Sc. degrees in industrial and production engineering and the Ph.D. degree from the Bangladesh University of Engineering and Technology, in 2009, 2013, and 2017, respectively. He is currently a Lecturer on system engineering and project management and also the Program Coordinator for Master of Decision Analytics and Master of Engineering Science with the School of Engineering and Information Technology, University of New South Wales (UNSW), Canberra, Australia. He is also the Group Leader of the Cross-Disciplinary Optimisation Under the Capability Context Research Team. He has written two book chapters and more than 95 technical journal and conference papers. His research interests include operations research, project management, supply chain management, artificial intelligence, cyber-physical systems, and information systems management. His research program has been funded by many organizations, such as the Department of Defence, Commonwealth Government, Australia.

MICHAEL J. RYAN (Senior Member, IEEE) received the bachelor's, master's, and the Doctor of Philosophy degrees in engineering. He is currently the Professor and the Director of the Capability Systems Centre (CSC), University of New South Wales (UNSW), Canberra. He has completed two years formal engineering management training in U.K. He has more than 35 years of experience in communications engineering, systems engineering, project management, and management. Since joining UNSW, he has lectured in a range of subjects, including communications and information systems, systems engineering, requirements engineering, and project management; and he regularly consults in those fields. He is the author/coauthor of 12 books, three book chapters, and more than 250 refereed journal and conference papers. He is a Fellow of Engineers Australia (FIEAust), a Chartered Professional Engineer (CPEng) in electrical and ITEE colleges, a Fellow of the International Council on Systems Engineering (INCOSE), and a Fellow of the Institute of Managers and Leaders (FIML).

FAISAL R. BADAL received the B.Sc. degree in mechatronics engineering from the Rajshahi University of Engineering and Technology (RUET). He is currently working as a Lecturer with the Department of Mechatronics Engineering, RUET. His research interests include smartgrid, artificial intelligence, machine learning, natural language processing, and robotics.

Z. TASNEEM (Member, IEEE) received the M.Sc. degree in electrical and electronic engineering from the Rajshahi University of Engineering and Technology (RUET), in October 2017. She joined RUET as a Lecturer in September 2015, where she is currently working as an Assistant Professor with the Department of Mechatronics Engineering. Her research interests include wind turbine aerodynamics, power electronics in renewable energy technology, and control systems.

MD H. AHAMED received the B.Sc. degree in mechatronics engineering from the Rajshahi University of Engineering and Technology (RUET), Bangladesh, where he is currently pursuing the M.Sc. degree in engineering with the Department of Computer Science and Engineering. He is currently working as a Lecturer with the Department of Mechatronics Engineering, RUET. His research interests include machine vision, artificial intelligence, machine learning, robotics, and image processing.

S. I. MOYEEN received the B.Sc. degree in computer science and engineering from the Rajshahi University of Engineering and Technology (RUET), Rajshahi, Bangladesh, where she is currently pursuing the M.Sc. degree with the Department of Computer Science and Engineering. She has worked as a Lecturer with the Department of Computer Science and Engineering, North Bengal International University, Rajshahi. She joined the Department of Mechatronics Engineering, RUET, as a Lecturer, in November 2019. Her research interests include machine learning, cloud computing, and the IoT, Web development, data mining, and big data, image processing, and artificial intelligence.

SAJAL K. DAS (Member, IEEE) received the Ph.D. degree from the University of New South Wales (UNSW), Australia. He has worked as a Research Engineer with the National University of Singapore (NUS), Singapore. He was a Visiting Academic with the University of Newcastle, Australia, and a Faculty Member with the Department of Electrical and Electronic Engineering, American International University, Bangladesh. He is currently the Head of the Department of Mechatronics Engineering, Rajshahi University of Engineering and Technology, Bangladesh, and the Director of the Control System Research Group, RUET. He is also the President of the Robotic Society of RUET and an Advisor of the Robotics and Automation Society, IEEE RUET SB, Bangladesh. His research interests include renewable energy generation and control, microgrid, smart grid, virtual power plant, cyber-security, and nano-positioning control. He serves as a Guest Editor for *IET Renewable Power Generation*, *Sustainability*, and *Energies*.

MD F. ALI is currently working as an Assistant Professor with the Department of Mechatronics Engineering, Rajshahi University of Engineering and Technology (RUET). His research interests include power electronics, control theory and applications, mechatronics, and artificial intelligence.

MD R. ISLAM (Member, IEEE) is currently pursuing the M.Sc. degree in mechanical engineering with the Rajshahi University of Engineering and Technology (RUET). He was appointed as a Lecturer with the Department of Mechanical Engineering, Bangladesh Army University of Science and Technology (BAUST), from February 2015 to August 2018. He is currently working as an Assistant Professor with the Department of Mechatronics Engineering, RUET. His research interests include mechatronics systems design, wind turbine aerodynamics, machine learning, and renewable energy.

DIP KUMAR SAHA (Member, IEEE) received the B.Sc. degree in mechanical engineering from the Rajshahi University of Engineering and Technology (RUET), in 2012, where he is currently pursuing the M.Sc. degree in mechanical engineering. He is currently working as an Assistant Professor with the Department of Mechatronics Engineering, RUET. After graduation, he joined Walton Hi-tech Industries Ltd., as a Research and Development Engineer. He served there above four years. In 2018, he joined the Department of Mechatronics Engineering, RUET, as a Lecturer. His research interests include mechatronics, robotics, machine learning, machinery condition monitoring, and failure analysis.

• • •