Academic Journal of Nawroz University (AJNU), Vol.12, No.3, 2023

https://doi.org/10.25007/ajnu.v12n3a1500

Visual-Based Simultaneous Localization and Mapping (VSLAM) Techniques for Robots: A Scientific Review

Fayez Saeed Faizi[1, 2], Ahmed Khorsheed Alsulaifanie[1]

[1] Department of Electrical and Computer Engineering. University of Duhok, Iraq-Kurdistan, Iraq
[2] Department of Energy Engineering. Duhok Polytechnic University, Iraq-Kurdistan, Iraq

ABSTRACT

The main problem facing autonomous robots is to navigate in an environment with the ability to determine its location and simultaneously build a map, SLAM technique can formulate this requirement efficiently. In this paper Filter-based, Graph-based, and AI-based Visual-SLAM techniques have been reviewed. The review shows that the first method suffers from high computations when the number of landmarks increases. The Graph-based algorithms are exposed to drift-error problems which cause a delocalization and require optimization. The AI-based vSLAM has the advantage of not-having complicated mathematical models in the algorithm, and it shows an efficient performance in various environments. The reviewed algorithms utilize different cameras including mono, stereo, and RGB-D cameras. The low-cost RGB-D cameras encourage implementation in modern autonomous robots. This work introduces a scientific-based overview of vSLAM to the reader, by explaining all phases of SLAM, the state-of-the-art algorithms, highlighting the strengths and weaknesses of each paradigm.

KEY WORDS: VSLAM, Localization, and Mapping, EKF-Based, Graph-Based, and AI-Based vSLAM.

1. Introduction

Robotics can be defined as the field that manipulates the surrounding environment physically via mechanical devices controlled through a computer. At the beginning of the 20th century, the term 'robot' was presented as science fiction in a play in 1917. Several years later it became true and started to take significant attention from both the scientific and industrial sectors. Currently, it has an essential role in the industry and draws a lot of attention from researchers in academic institutions. Today, robots cover a wide range of devices and systems, starting from the simplest manipulator that does basic tasks, extending to the most complex autonomous robots (Thrun, et al., 2005; Hockstein, et al., 2007).

The studies related to robots are not totally new science, but they can be considered a combination of various engineering domains such as; mechanical engineering which deals with moving parts, and electrical engineering, which is responsible for powering the robot and sensing the environment. Then, the computer engineering side is the brain that analyzes the data and makes decisions that control the robot (Wallén, 2008). The early robots were fully controlled and programmed to perform specifically dedicated tasks. However, in the following decade, fundamental steps were taken toward building a robot that can analyze data and make decisions based on that led to the born of autonomous robots (Bensalem, et al., 2008).

An autonomous robot is defined as a robot that can do a specific task(s) based on the decisions it takes without the intervention of humans (Khairuddin, et al., 2015). The process of 'self-controlling' the robot is complex and faces several challenges. In a Mobile Robot case, the autonomous robot may move in an uneven environment with different obstacles that it should avoid colliding with. The challenge is more complex if the robot needs to move within an unknown environment without having any previous information. This means that it should be able to know its location as well as build its own map that will be used to navigate the area, in addition to obstacle avoidance task (Thrun, et al., 2005; Siegwart and Nourbakhsh, 2004). Figure (1) shows the block diagram of a mobile self-exploring robot process steps.

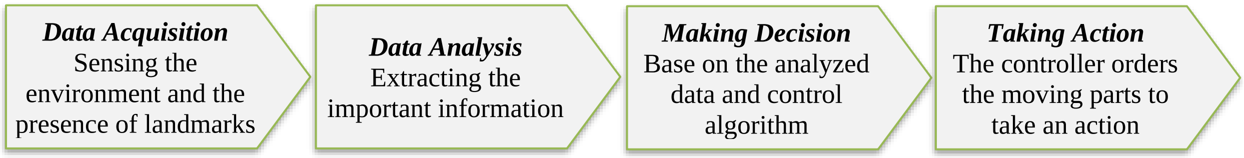

Figure (1) Block diagram of autonomous robot

For a self-exploring robot, it's crucial to 'know' its location and build an environment map accordingly. In order to build a map, the robot needs to identify its current location. However, the robot needs a valid map its surrounding to relatively identify its location; for new environment, this could lead to the dilemma of 'egg or chicken first' (Birk and Pfingsthorn, 2016).

The mentioned development takes some time and a lot of effort to resolve this problem. Several researchers studied this issue and tried to find a solution for this issue. The real solution for the 'map or location' dilemma was presented by Leonard and Durrant-Whyte (Leonard and Durrant-Whyte, 1991) and finalized in (Smith and Cheeseman, 1986), where they build an algorithm that allows the robot to know its location and at the same time creates a map for the environment that the robot work in it. Their work was based on the method presented in (Durrant-Whyte, et al., 1995) and utilized the Extended Kalman Filter (EKF) method. This method can be considered as the first Simultaneous Localization and Mapping (SLAM) algorithm that solves the problem of building a map and defining robot location (Alsadik and Karam, 2021).

After solving the SLAM problem, different improvements and new algorithms were proposed to overcome some difficulties related to SLAM, such as complexity; certainty problem; analysis and action time required; accuracy; and so on. From a data-analysis point of view, SLAM can be categorized into three main types: Filter-based, Smoothing technique-based, and AI-based SLAM (Saeedi, et al., 2016). Each one of these methods has its pros and cons; these methods will be explained in some detail in section II. Also, there is another classification for SLAM based on map representation, such as feature-based, view-based, polygon-based, and appearances-based SLAM. Both classification types are known, but the first taxonomy is mainly used (Correll, 2016).

Section II explains the SLAM methodology whereas section III presents the implementation of the SLAM technique with examples of each method. The following section introduces the vSLAM and shows some of its virtues and characteristics. While section V presents the latest works on vSLAM. Finally, in section VI, this work's conclusion is discussed.

2. Simultaneous Localization and Mapping (SLAM)

SLAM can be defined as the ability of the mobile robot to determine its location and at the same time, gradually, build the map of the operation area. There are different types of SLAM systems, but all of them share the same general structure that consists of two essential parts: the Front End and the Back End. The system collects the data via its sensor(s) in the front-end phase, extracting features and tracking duly. As the robot moves, more data are collected, and the successive incoming data (then extracted features from it) are gathered and combined with its associates, this process is known as Data association (Alsadik and Karam, 2021; Correll, 2016). The technique that deals with this issue is explained in detail in the coming sections.

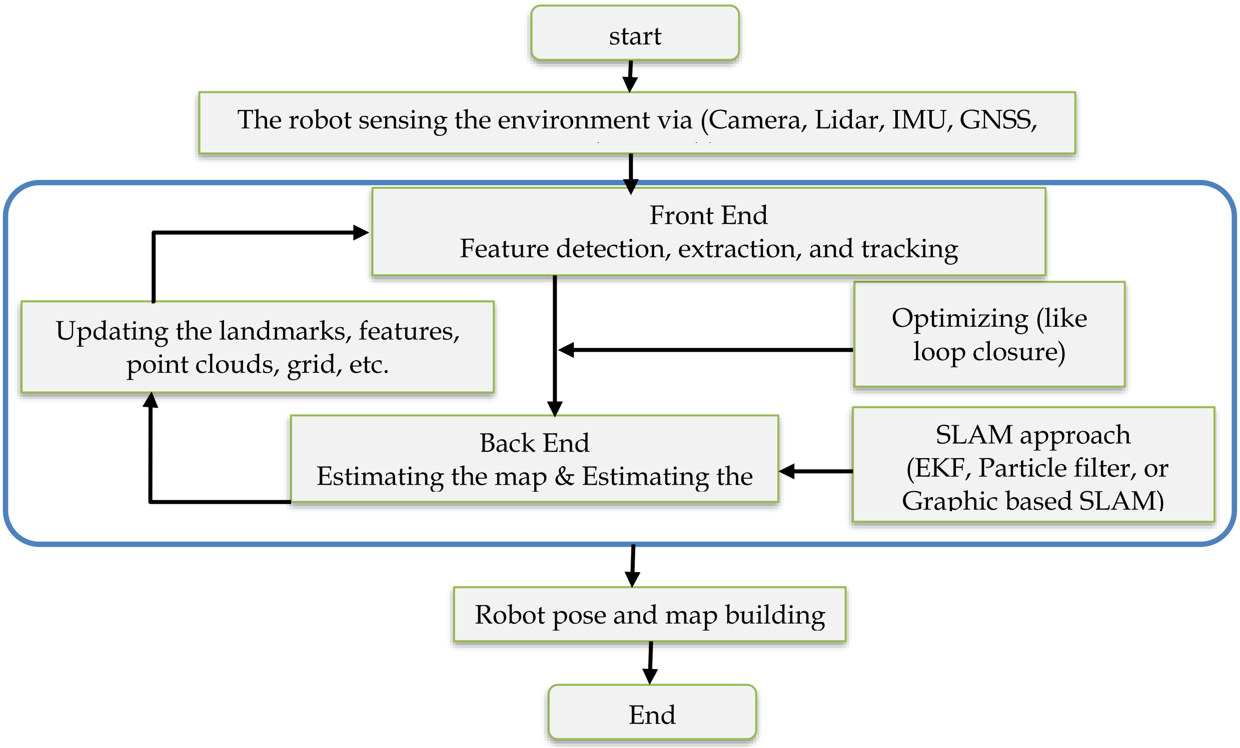

Figure (2) The general form of the SLAM technique

The back-end refers to the step of building the map and pose estimation where these processes are based on complex mathematical methods such as filtering, like Extended Kalman Filter (EKF) and particle filter, or it is based on smoothing technique like Graph-based SLAM optimization. Recently, the SLAM paradigm has been built by utilizing Artificial Intelligence (AI). The SLAM process mainly consists of estimation and update strides (Jiménez, et al., 2016). Figure (2) shows the flow chart for the SLAM techniques, both front-end and back-end.

High certainty must be available for SLAM to localize the robot accurately and to build the map. In contrast, to real-life case, the robot faces high uncertainty in the environment and with almost no knowledge about its position at the beginning of its operation. To overcome this issue, the robot deploys probabilistic tools to reduce the uncertainty after each iteration (movement) as it sees new landmarks (Mur-Artal, et al., 2015). The models that deal with the uncertainty problem, in specific, and the SLAM technique, in general, are explained in detail in the next section.

3. Implementing SLAM

As aforementioned, from a broad point of view, there are two types of SLAM techniques: filtering-based and smoothing-based. The most popular one is the filtering techniques used in conjunction with the Extended Kalman Filter (EKF) (Sorenson, 1966) and particle filters (Arulampalam, et al., 2002). On the other hand, the most common smoothing approach is the graph-based SLAM (Grisetti, et al., 2011). Each one of the mentioned methods has its advantages and disadvantages that will be discussed.

Suppose that, there is a robot that moves in an unknown environment that can observe the landmarks via a sensor attached to the robot. The following parameters are considered:

x_k: A vector refers to the robot's pose (location and orientation).

u_k: Represent the control vector deployed at the time (k-1) to move the robot to state x_k.

m_i: A vector that indicates the location of landmark (i).

$z_{i,k}$: The observation measured via robot of the landmark (i) location at the time (k).

At the beginning, robot controls (U1:T = {u1,u2,u3,…..,uT}) and sensor observations (Z1:T = {z1,z2,z3,….,zT}) are known. While the map of the environment (m) and robot's path (X0:T= {x0,x1,x2,…..,xT}) should be determined. For the probabilistic SLAM, the probability distribution needs to be calculated for the robot's pose and the map gave the control, the observation, and the initial state of the robot.

$$P(x_k;\mathrm{m} \mid z_{0:k}, u_{0:k}, x_0) \qquad (1)$$

The SLAM problem can be solved in iterative ways, where the calculation needs that (x_k) and (z_k) are known to explain the effect of actual observations and incoming control. The probability for the robot to sense (perform an observation) is described by the model:

$$\mathrm{P}(z_k \mid x_k, \mathrm{m}) \qquad (2)$$

The robot's motion can be defined as a probability distribution given the previous state and the command control only, independent of the map and sensor observations.

$$P(x_k \mid x_{k-1} \; u_k) \qquad (3)$$

Solving equation (1) leads to the two steps for solving the SLAM problem: the prediction step and the correction step. The first step is estimated as:

$$P(x_k,m \mid Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k \mid x_{k-1}, u_k) * P(x_{k-1},m \mid Z_{0:k-1}, U_{0:k}, x_0)\, dx_{k-1} \qquad (4)$$

While the second step is in the form of:

$$P(x_k,m \mid Z_{0:k}, U_{0:k}, x_0) = \frac{\mathrm{P}(z_k \mid x_k,\mathrm{m})\, P(x_k,m \mid Z_{0:k-1}, U_{0:k}, x_0)}{\mathrm{P}(z_k \mid Z_{0:k-1}, U_{0:k})} \qquad (5)$$

The last two equations show the main iterative procedure required to determine the robot's location and the environment's map by utilizing the sensed observations (z_(0:k)) and input control (u_(0:k)). To solve equations (4) and (5) efficiently, a suitable representation for observation equation (eq. 2) and motion equation (eq. 3) are required. In the literature, several approaches are proposed; the Extended Kalman Filter (EKF), Graph-Based SLAM, and AI-based SLAM. The mathematical bases for each method are explained in detail together with relevant SLAM implementation examples.

3.1 Extended Kalman Filter (EKF) SLAM

The first solution for the SLAM problem was based on the Kalman filter technique (Leonard and H. F. Durrant-Whyte, 1991). The Kalman filter is a recursive Bayes filter that can be split into two steps; the prediction step is then followed by the update step (Durrant-Whyte, et al., 1995). The Inertial Measurement Unit (IMU) is used in the first step to predicting the robot's motion. On the other hand, the data collected by the camera, a LiDAR, or other sensors are used (after extracting the important features) in the prediction step (Mohamed, et al., 2019; Karam, et al., 2019).

The EKF SLAM consists of 5 phases:

- **State prediction**: Apply the control and estimate the new position of the robot given the control.
- **Measurement prediction**: What is expected to be observed given the best estimate for the robot.
- **Measurement:** Take the actual measures.
- **Data association**: Which landmark does the current observation, actually, corresponds to, then calculate the difference (deviation) between expected observation and obtained ones.
- **Update phase**: Updates the matrices and repeats the process.

These steps are repeated as a cycle to determine the final location and the environment map. The process starts by representing the motion of the robot; from equation (3), we get:

$$x_k = \mathrm{f}\,(x_{k-1}, u_k) + w_k \qquad (6)$$

Where: f(x) represents the kinematic model of the robot, and (w_k) is a Gaussian zero-mean motion turmoil additive, which has a covariance (Q_k). Then and from equation (2), the observation model can be described in the form of:

$$\mathrm{z(k)} = \mathrm{h}\,(x_k, \mathrm{m}) + v_k \qquad (7)$$

Where: h(x) represents the observation's geometry, and (v_k) is a Gaussian uncorrelated error of the observation that has a zero mean and covariance (R_k). The mean can be calculated by utilizing these definitions and applying the standard EKF (Maybeck, et al., 1979). The resulting equation will be:

$$\binom{\hat{x}_{k|k}}{\hat{m}_k} = E\left[\begin{smallmatrix} xk \\ m \end{smallmatrix} \mid Z_{0:k}\right] \quad (8)$$

And the covariance matrix will be:

$$P_{k|k} = \begin{bmatrix} P_{xx} & P_{xm} \\ P_{xm}^T & P_{mm} \end{bmatrix}_{k|k} \quad (9)$$

The prediction step can be described in the form of:

$$\hat{X}_{k|k-1} = f(\hat{X}_{k-1|k-1}, U_k) \quad (10)$$

$$P_{xx,k|k-1} = \blacktriangledown \mathrm{f}\, P_{xx,k-1|k-1} \blacktriangledown \mathrm{f}^T + Q_k \quad (11)$$

Where: ▼f is the Jacobian matrix of (f). Applying the last two equations will compute the parameters of the prediction step, then the robot will update these parameters by using:

$$\binom{\hat{x}_{k|k}}{\hat{m}_k} = \binom{\hat{x}_{k|k-1}}{\hat{m}_{k-1}} + W_k\left[z(k) - h(\hat{x}_{k|k-1}, \hat{m}_{k-1})\right]$$

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^T$$

Where:

$$S_k = \blacktriangledown \mathrm{h}\, P_{k|k-1} \blacktriangledown h^T + R_k$$

$$S_k = P_{k|k-1} \blacktriangledown h^T S_k^{-1}$$

The results is fed back as (x_(k-1)) to the first step, the process is repeated, all over again, until the robot navigates the whole environment and reaches the beginning 'point'. Then 'the loop closure' technique may be required to complete the map building process.

The EKF SLAM technique suffers from some shortcomings that make it unsuitable in some applications. The high computation burden is clear, where it needs to update all covariance matrices each time it makes a new observation (Whyte and Bailey, 2006). Furthermore, the EKF SLAM deploys a linearized scheme of nonlinear models for both observation and motion, which could lead to conflicting results that affects the whole SLAM process (Julier and Uhlmann, 2001). Figure (3) demonstrates the entire procedure of EKF SLAM.

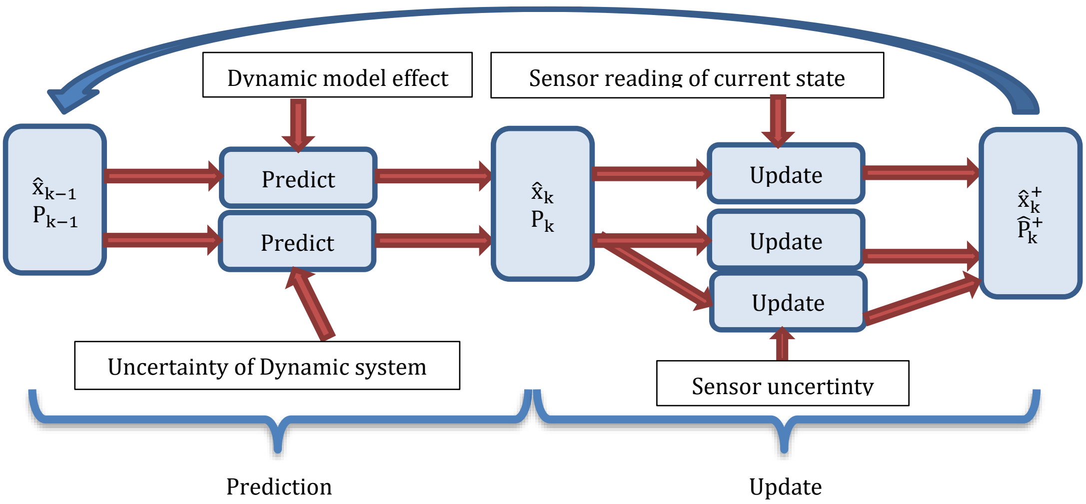

Figure (3) The steps of EKF SLAM

Figure (3) shows that the EKF SLAM scheme is an online SLAM technique, where only the next state is estimated, not the whole trajectory (Alsadik and Karam, 2021).

3.2 Graph-Based SLAM

The basic form of the graph-based SLAM was first presented in 1997 by Lu and Milios (Lu and MILIOS, 1997). It took many years before it became popular due to its complexity in solving the error-minimization issue via traditional models. This method is known as a Full SLAM technique, as it estimates the whole trajectory of the mobile robot based on the available measurements. The graph-based SLAM scheme is, generally, based on a least-square error minimization (Dellaert and Kaess, 2006).

The beginning of solving the SLAM problem via graph-based technique is building a graph consisting of different connected nodes. These nodes are referred to as the sensed (measured) landmarks or robot poses, while the edge between two nodes represents sensor measurement which constrains every two related poses. It's worth mentioning that the constraints may be antithetical due to the noise that affects the observations. After building the graph, the challenge is to find the optimal configuration of the nodes that give the best harmony with the available measurements. This means that a large error minimization issue needs to be solved. It's clear now that the graph-based SLAM consists of two phases: the graph construction phase, then the graph optimization phase (Grisetti, et al., 2011; Olson, et al., 2006).

To make the process clearer, let x= $[x_1, x_2, \ldots, x_T]^T$ refer to a vector of parameters, where x_i represents the pose of node (i), and $z_{i,j}$ describe the mean of the measurements between node (i) and node (j). Also, $\hat{z}_{i,j}(x_i, x_j)$ is describing the prediction measurements between node (i) and node (j). Now, assume $e_{i,j}(x_i, x_j)$ is a function that determines the variation between the actual observation sensed by the robot and the prediction observation. This means that:

$$e_{i,j}(x_i, x_j) = z_{i,j} - \hat{z}_{i,j}(x_i, x_j) \qquad (12)$$

The log-likelihood $l_{i,j}$ of the the $z_{i,j}$ is given by:

$$l_{i,j} \propto [z_{i,j} - \hat{z}_{i,j}(x_i, x_j)]^T \, \Omega_{i,j} \, [z_{i,j} - \hat{z}_{i,j}(x_i, x_j)] \qquad (13)$$

Where $\Omega_{i,j}$ represent the information matrix of the measurements between node (i) and node (j). Figure (4) shows the parameters that be utilized to define the edge of a graph.

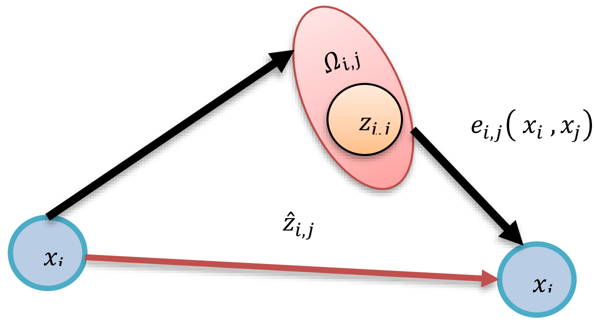

Figure (4): Aspects of the graphs' edge

Equation (13) indicates that a configuration of the nodes that minimizes the negative log-likelihood F(x) for all observations can be obtained by using the maximizing likelihood scheme.

$$F(x) = \sum_{<i,j> \in C} e_{i,j}^T \, \Omega_{i,j} \, e_{i,j} \qquad (14)$$

$$x^* = argmin \, F(x) \qquad (15)$$

Assume that the initial guess pose ($\breve{x}$) is known, then equation (15) can be solved numerically via standers optimization techniques like the Gauss-Newton algorithm. From Taylor expansion of the error function (equation 12), the solution is approximated around the initial guess ($\breve{x}$).

$$e_{i,j}(\breve{x}_i + \Delta x_i, \breve{x}_j + \Delta x_j) = e_{i,j}(\breve{x} + \Delta x) \qquad (16)$$

$$\approx e_{i,j} + J_{i,j} \, \Delta x \qquad (17)$$

Where, $J_{i,j}$ represent the Jacobian of $e_{i,j}$(x) vector calculated in $\breve{x}$. The expression showed in equation (14) can be written after substituting equation (17) in the error term.

$$F_{i,j}(\breve{x} + \Delta x) = e_{i,j}(\breve{x} + \Delta x)^T \, \Omega_{i,j} \, e_{i,j}(\breve{x} + \Delta x) \qquad (18)$$

$$\approx (e_{i,j} + J_{i,j} \, \Delta x) \, \Omega_{i,j} \, (e_{i,j} + J_{i,j} \, \Delta x) \qquad (19)$$

$$= e_{i,j}^T \, \Omega_{i,j} \, e_{i,j} + 2 e_{i,j}^T \, \Omega_{i,j} J_{i,j} \, \Delta x + \Delta x^T \, {J_{i,j}}^T \Omega_{i,j} J_{i,j} \, \Delta x \qquad (20)$$

$$= c_{i,j} + 2 b_{i,j} \, \Delta x + \Delta x^T H_{i,j} \Delta x \qquad (21)$$

With the previous approximation, equation (14) can be written as:

$$F(\breve{x} + \Delta x) = \sum_{<i,j> \in C} F_{i,j}(\breve{x} + \Delta x) \qquad (22)$$

$$\approx \sum_{<i,j> \in C} c_{i,j} + 2b_{i,j}\, \Delta x + \Delta x^T H_{i,j} \Delta x \qquad (23)$$

$$= c + 2b\, \Delta x + \Delta x^T H \Delta x \qquad (24)$$

Equation (24) can be seen as a quadratic form which can be minimized in (Δx) through solving the linear system:

$$H\, \Delta x^* = -b \qquad (25)$$

Where (H) is the information matrix as it was acquired through projecting the error via the Jacobians. The linear system is shown in equation (25) can be solved via sparse Cholesky factorization (Grisetti, et al., 2011; Davis, 2006). Finally, the initial guess is added to the calculated increments to determine the linearized solution:

$$x^* = \breve{x} + \Delta x \qquad (26)$$

The well-known Gauss-Newton iterative algorithm is used to perform the linearization in equation (24), finding the results from equation (25), and update step in equation (26).

After showing the details of both types, it's clear that these methods have complex computations that require high computational devices such as parallel processing devices or multi-core CPUs. This is considered a shortcoming in the robotics field, as hese devices are expensive, and most microcomputers do not support such high-performance equipment. That will affect calculation time and the accuracy of the SLAM (MathWorks, 2021).

3.3 AI-Based SLAM

In the last few years, Artificial Intelligence (AI) has been used to solve the full SLAM problems to replace the conventional odometry paradigm. At the beginning of applying deep learning in solving the SLAM problem, it was focused on localization only via visual odometry, without considering the mapping issue. Then it was developed to solve the full SLAM problem. The main benefit of utilizing AI in SLAM is preserving the system's high performance in complex environments and reducing the percentage of the wrong estimation in visual SLAM (Khairuddin, et al., 2015; Alsadik and S. Karam, 2021). AI-based visual SLAM can extract the inter-frame pose of two successive captured images from a mobile robot. Also, AI-based SLAM can efficiently estimate the rotation and translation of the camera (six degrees of freedom 6DoF), and it can predict the depth distance of the objects in an image captured by a single camera. Both Supervised and Unsupervised methods are used in the SLAM problem for the mono camera, RGB-D camera, and stereo cameras based robots (Khairuddin, et al., 2015; Saeedi,et, al., 2016). Table (1) shows a general comparison among the mentioned three ways, indicating the virtues and short-comes of each type.

Table (1): Different SLAM method comparison

#	Method	Explanation	Pros	Cons
1	EKF-Based SLAM	Utilizing the idea of the Kalman filter to solve the SLAM problem.	• Good performance if features are distinct. • Efficient online and full SLAM. • Parametrization is not required.	• Unable to identify the absence of the feature. • The processing time will quadratically increase as a new feature is added to the state space.
2	Graph-Based SLAM	The smoothing paradigm is utilized in map and trajectory estimation.	• Complete trajectory is updated. • Appropriate for large-scale environments	• Created map needs to be adjusted. • Need high computation cost.
3	AI-Based SLAM	Based on artificial intelligence	• Mathematical models were not required. • Efficient performance.	• Need parameter tuning and training. • Time-consuming during the training process.

4. Visual Simultaneously Localization and Mapping (vSLAM)

The visual SLAM system refers to the mobile robot that uses an optical sensor (camera) to sense the environment, and the incoming data for the SLAM technology (the input data) is the taken images (Alsadik and S. Karam, 2021). There are different types of visual-based SLAM that have a variety of characteristics related to each class. For instance, the Feature-based vSLAM, the Direct vSLAM, and the RGB-D camera vSLAM scheme (Taketomi et al., 2017).

One of the Feature-based vSLAM techniques is MonoSLAM, presented by Davison et al. in 2003 (Davison, 2003). They used the EKF paradigm to localize a robot and build a 3D map using a single camera with a 6 Degree of freedom (DoF) attached to the robot. On the other hand, the DTAM scheme was proposed by (Newcombe et al., 2011) as a realization of the Direct vSLAM method. At the same time, the RGB-D-based vSLAM robots use (usually) one camera that provides information of the fourth matrix (the depth matrix D) to estimate the distance of the near obstacles. But, this type uses techniques and algorithms different from Feature-based vSLAM, which will be clear in the next section (Taketomi et al., 2017).

Overall, the vSLAM faces more technical difficulties than other input sensors (such as 360° laser sensor) because of the limited field of view and higher computational cost required, making it slower. In addition, the vSLAM is sensitive to the variance in the light in the operation environment, which may affect the robot's performance accuracy in the case of significant illuminance change. Nevertheless, the vSLAM technique is preferred to be used due to its low cost, and this technique is also required in applications that need passive sensing (Alsadik and S. Karam, 2021; Muhammed, et al., 2009).

5. RELATED WORKS

Since the first SLAM algorithm was presented, several researchers have tried to improve the standards paradigm and propose new and more efficient algorithms. After showing the idea of the vSLAM and explaining the mathematics behind it, some works related to this topic will now be displayed. Nevertheless, the strengths and weaknesses of each proposed method is discussed with summaries in table (2).

(Jajulwar and Deshmukh, 2013) proposed an algorithm that tries to improve the performance of standard EKF SLAM by using two encoders and an image correlation method. The system has three phases: capturing the images; calculating robot location via the two encoders; and utilizing the image correlation paradigm to implement the map used in pose estimation, and combining all that via distribution filter to reduce the noise and complete the SLAM operation. The authors claimed that their algorithm improves the overall performance of robot navigation tasks.

(Makhubela et al., 2019) presented a framework for implementing vSLAM to solve the environment's light intensity issue by using an image filtering algorithm. The proposed scheme consists of five layers: Capturing the images; Pre-processing; Applying a Light Filtering Algorithm; positioning the robot and building the map; and navigating the robot. The image filtering reduces the effect of light intensity noise coming from the input image and allows the robot to operate in noisy environments. The proposed framework is built over the EKF SLAM for localization and mapping, jointly, with the A* algorithm to support robot's navigation. Simulated results showed that this paradigm reduced the Root Mean Squared Error (RMSE) to 0.13, and the robot could navigate in a noisy workplace.

(Dib et al., 2014) built a new algorithm for the vSLAM technique. The paradigm utilized the Chamfer distance approach to estimate the pose and used the occupancy grid to help in creating the map. The proposed method did deploy feature matching but, it minimizes the space between the occupancy grid and the feature points via distance map. This means that this approach overcomes the high computation issue of feature matching and the wrong matching problem that impacts the motion accuracy. The presented algorithm is designed, mainly, for the RGB-D camera; nevertheless, due to the use of the occupancy grid, this method can be deployed for other sensor types as well, such as telemeter sensors.

(Benavidez et al., 2014) introduced in the same year a scheme that also tries to reduce the memory requirement and computation time in vSLAM by avoiding feature matching. To achieve these goals, the proposed algorithm deploys the rg-Chromaticity paradigm, which depends on the RGB concentrations for the feature detection step. Then the extracted features will be matched with the available dataset before it goes to the parallel processing system that utilizes a voting mechanism for matching. Experiments results show that the proposed model can identify the matching for a cropped image of size equal to ¼ of the original image size, which means a reduction in storing space reaches 75% of the original size. The smaller image and dataset size lead to faster matching and drop in the required time.

(Alismail, et al., 2016) worked on improving the Bundle Adjustment (BA) to improve the accuracy in the vSLAM application. The presented paradigm was based on maximizing the photometric consistency instead of minimizing the reprojection error via tracked feature to compute the optimal parameters. Results show an increase in navigation accuracy through minimizing photometric error via different scenes, compared to the results obtained from minimizing the reprojection error. The presented model is correspondence's independent, which makes it applicable in any non-vanishing gradient pixel image, and it is appropriate for both; indoor and outdoor environments.

(Artal, et al., 2015) merged several techniques to build a novel feature-based monocular vSLAM algorithm named ORB-SLAM. The proposed system consists of Place recognition, loop closing via scale-aware, and utilizing the co-visibility data for large operation environments. The main idea behind the ORB SLAM is that the feature used for loop closing and place recognition to determine localization are the same features used by the tracking and mapping technique. Experimental results showed that the ROB SLAM could operate in a large environment in real-time and perform a real-time loop closing via pose graph optimization. The results, also, shows the ability of the system to automatically and robustly complete the initialization process.

(Artal, et al. 2017) introduced ORB-SLAM2, the improved version of the previously presented ORB-SLAM. The main update is that the new version can use not only the monocular camera but also supports stereo and RGB-D cameras. The proposed system is a feature-based algorithm, where it will pre-process the incoming data (input images) to extract the features related to the prominent key-point places. Then, the input images will be deleted, and the model will utilize the extracted features for the remaining operations. That is why the proposed system is independent of the input vision sensor type. The ORB-SLAM2 as a feature-based method showed better results than a Direct mode and required less computation cost. Nevertheless, the proposed algorithm has a lightweight localization model that provides a robust localization with zero drift in known places.

(Campos, et al., 2021) investigated ORB-SLAM, ORB-SLAM2, and ORBSLAM Visual-Inertial to build a new system named ORB-SLAM3. The presented system was the first system capable of taking complete advantage of short-term, mid-term, and long-term data association that leads to zero drift for the known places. The created map is used whenever needed to perform precise localization; this can be done by utilizing multi-map association performed that help the system to match the Bundle Adjustment elements from old mapping tour. Experimental results showed that the proposed method overcomes the previously presented algorithms in accuracy, robustness, and multi-mapping.

(Pumarola, et al., 2017) try to solve the problem of low texture related to the SLAM technique in environments via presenting the PL-SLAM approach. They proposed an algorithm based on line correspondences and built over ORB-SLAM, where the system will modify the ORB-SLAM pipeline to perform three steps: Tracking, Local mapping, and Loop closing. The Pl-SLAM can operate efficiently when the larger amount of the points of the incoming image are died out, and it will be initialized by utilizing the line correspondences detection from three sequential frames. The proposed method can also perform line-based map initialization that uses five-line correspondences from the sequential three images to predict a 3D map of the environment. The authors claim that their paradigm overcomes the performance of ORB-SLAM in low-textured environments in terms of accuracy and efficiency.

(Wang, 2018) proposed an algorithm using a Convolutional Neural Network (CNN) to build a SLAM system based on Artificial Intelligent (AI). The presented model utilizes parallel multi convolutional layers instead of using a single layer, and it also deploys a parallel pooling method to reduce the size of the model parameter. Then, the fully connected layer will process the coming signal (after filtering and pooling) to build a global map that will pass to the robot's navigation. The results of testing this algorithm indicate the superior of this method over the traditional CNN method in reducing the overall error.

(Vincent, et al., 2020) combined the Extended Kalman filter and Deep Learning to present a fast model for improving the vSLAM in dynamic environments. After getting the input images, the model will identify the observed objects based on the prior information about the dynamic objects. The trained neural network will perform paradigm segmentation to define the class of the observed object. Each identified dynamic object instance has a Dynamic Object State (DOS) containing its bounding box, binary mask, and object type. Then, the Masked Depth Image (MDI) is built by applying the DOS to the original RGB-D image. The tracking module can estimate the speed of each object after determining the masked object's 3D centroid. Finally, the Moving Object Classification Module (MOC) identifies the objects as dynamic or idle depending on their class and predicts shape deformation and the object's speed. Testing results of the proposed algorithm showed the prior of the algorithm compared to the available paradigms. Also, it indicates the fast operations of this algorithm and that it can be executed on a robot moving at a medium velocity.

(Ai, et al., 2020) provide a new scheme for vSLAM in dynamic environments the operate efficiently and solve the outlier problem. The proposed system consists of five main steps. In the first step, the incoming row image will be processed, then in the tracking (second step), keyframes from the consecutive images will be extracted. The object detection step will feed the keyframes to a convolutional neural network that will detect the object based on its training ("understanding" is has). In the fourth step, the Dynamic Object Probability (DOP) will be used to recognize the static and dynamic areas and the motion of dynamic probability via point matching expansion and feature matching. The map will be built at the last step, loop closing will be determined, and a full BA deployed. Results showed that the proposed system has high accuracy and high stability in dynamic environments. As well as, it decreases the drift and tracking error and improves the robustness compared to ORB-SLAM2.

(Bruno and Colombini, 2021) merged the Deep Neural Network (DNN) feature extracting technique known as the Learned Invariant Feature Transform (LIFT) with the traditional geometry-based SLAM (ORB-SLAM) to build a new monocular vSLAM named LEFT-SLAM. The supervised DNN will go throw three steps to perform a robust feature detection: local feature detection step, orientation estimation, and the description step. The gotten data will be feed to the geometry-based SLAM for completing the localization and map building. Unlike traditional ORB-SLAM, our algorithm performs the map building step in a sequential way after the tracking phase rather than parallel, while the loop closing task remains to be done in parallel. The system will construct two types of maps: a keyframe-based map and a graph-based map, which help deploy the bundle adjustment optimization to predict the poses. Experimental results showed that using DNN improves the performance of vSLAM and makes it more robust than traditional techniques, and it can work in indoor and outdoor environments.

(Li, et al.,2021) introduced a new real-time vSLAM algorithm based on deep learning and deployed a multi-task feature extraction network with unsupervised feature points. The system utilizes a CNN to discover feature points and a descriptor instead of a conventional feature extractor. The algorithm is based on ORB-SLAM2, but the original matching paradigm is disabled and uses the nearest neighbor matching technique. After that, the reprojection error is minimized via nonlinear optimization to compute the pose of the camera. Results indicate that this method can work in low texture environments without an efficient drop, and both stability and precision of the system were improved.

(Maxime, et al., 2021)present the OV2 SLAM a totally online algorithm. The proposed system can work with both monocular and stereo cameras; it covers a wide range of frame rates and builds different map sizes. It's based on the idea of the multi-layer structure, considering the critical and non-critical tasks. The introduced method started with improving the contract of the incoming image via the CLAHE technique as pre-processing image phase. The second phase will utilize the guided coarse-to-fine optical flow model as a key-point tracking. Then the outliers will be eliminated by applying the RANSAC method. In the fourth phase, the pose estimation process will be obtained, where the 3D key-points projection error will be minimized via the Huber coast model. In the end, new keyframes will be created by the front-end-thread. OV2 SLAM shows better results in real-time operation and accuracy than several other algorithms in their literature. Also, it indicates that the proposed paradigm can be applied in the ground and aerial robots for indoor and outdoor environments.

(Chen, et al., 2021) utilized the concepts of Deep Learning to introduce a new visual place recognition and feature detection algorithm. Considering the Euclidean space, the distance constrain is optimized via a multi-constrain loss function. This can be done by thoroughly analyzing multiple constraints related to the spacing rapport in the visual spot recognition issue. The proposed algorithm makes the overall calculation burden depend on the training places instead of the number of images used in the training phase. The proposed algorithm supports all types of CNN to extract the required features. The authors claim that their system outperforms better than other traditional algorithms from efficiency and effectiveness points of view. It is worth mentioning that the presented method can work in complex environments with dynamic changes.

(Zhou, et al., 2021) worked on improving the standard ORB-SLAM by combining point and line features with it. The proposed method was based on the idea of using points and lines features to increase the accuracy of the extracted features. The system consists of three steps: Tracking, Local mapping, and Loop closing. In the first step and via line and points processing, the features will be extracted, followed by pose estimation and local map tracking. In a second step, the map will be built and then updated in points and lines terms, then a BA optimization through lines and points will be performed. In the third step, loop closing will be done in three threads: loop closure detection, loop closure correction, and global BA optimization. Testing the proposed algorithm shows that it can compete with the available algorithms, indicating accurate and robust operation and decreasing drift and losses.

(Liu and Miura, 2021) introduced a novel vSLAM method named RDMO-SLAM that benefits from semantic information more efficiently while keeping the real-time operation by utilizing dense optical flow via inserting semantic label estimation. The presented scheme uses a modified version of RDS-SLAM and ORB-SLAM3 and adds two additional steps, optical flow and velocity prediction. This will make the proposed algorithm work efficiently in a dynamic environment with the real-time process and overcome the slow speed of Mask R-CNN segmentation. Also, velocity constrain was added via predicting landmarks' velocity by utilizing optical flow, which will decrease the effect of dynamic objects. The landmarks are classified into three classes, dynamic, static, and unknowns, depending on movements probability calculated previously. The results indicate that the proposed method effectively works in a dynamic environment while keeping the real-time operation with heavy segmentation ways.

(Liu and Miura, 2021) proposed a KMOP-SLAM algorithm that neglects the outliers of the dynamic obstacles during the tracking process, which will make the vSLAM more robust in dynamic environments. By utilizing the power of human detection and unsupervised learning, the drift error of tracking is reduced. The presented system consists of a person detection model, unsupervised learning segmentation, geometric check, and dynamic object detection algorithm based on probability. Feature detection, human detection, and segmentation modules are working in parallel. The RGB image goes through the human detection system for human finding. Then it is fed to feature detection for ORB feature extraction used in moving object detection and pose prediction. At the same time, k-means will be applied to the depth image to calculate the pixel label. Now, dynamic obstacles are defined via the human detection module and geometric constraints. In the end, the camera's pose will be estimated via rigid features after outliers have been removed. Examining the presented algorithm shows that the outliers features related to dynamic obstacles were detected and eliminated efficiently, resulting in more robust vSLAM with minor drift error and good tracking performance.

(Zhang, et al., 2021) combined the deep learning technique with the geometry constraints of the traditional stereo Visual Odometry (VO) system to implement unsupervised pose correction for stereo VO-based vSLAM paradigms. The proposed module process the output of the conventional VO via CNN, where the images from both cameras are analyzed through an encoder-decoder network paradigm. The pose correction step didn't rely on the six-DoF dataset ground of the truth. The system will create an explainability mask and a depth map. The introduced network retracts a pose correction, leading to positioning error due to the violation of modeling assumptions to make the traditional stereo VO precise. Experimental results indicate the accuracy of the proposed method in terms of positioning accuracy, reducing the error drift.

Table (2): Related works summary

Ref.	Method Based on:	Camera Used	Contribution	Advantages	Disadvantages
Jajulwar and	Distributed filter	Mono-Camera	Sensor errors and estimating the pose were	• Improve the navigation	The map constructed matched the

Deshmukh, 2013			performed by utilizing distributed filter approach on "image matching points".	performance of the robot. The model is applicable for linear and nonlinear measurements.	environment by a ratio of 28.74, which is a low ratio.
Makhubela et al., 2019	Filtering technique combined with A* algorithm	Mono-Camera	The Light Filtering Algorithm was deployed to address the light intensity challenge.	• Work in noisy environments. Reduced the RMSE to a minimum of 0.13	• Suitable for static environments only. As it combines EKF and A*, it suffers from high computation cost.
Dib et al., 2014	Chamfer Distance and Occupancy Grid	RGB-D camera, but can be applied to other sensors like telemeter	Avoid feature matching, and instead, it minimizes the space between the occupancy grid and the feature points via a distance map.	• Avoid expensive operations (which make it fast) It applies to different types of sensors.	As it uses the distance between feature points, it is suitable for static environments only
Benavidez et al., 2014	rg-Chromaticity paradigm and parallel processing	RGB-D camera	Eschew a full feature matching, and deploy the rg-Chromaticity algorithm via parallel processing on the cropped image.	• Reduce the storage memory required. Minimize the processing time needs.	It can be deployed for static environments only
Alismail et al., 2016	Bundle Adjustment (BA) without correspondences	Mono-Camera	maximizing the photometric consistency and assessing the posts implicitly.	• It is applicable in indoor and outdoor environments. Showed better accuracy compared to traditional BA.	The initial guess of the camera pose is needed by geometric
Artal, et al., 2015	Merging bundle adjustment, place recognition, loop closing via scale-aware, and utilizing the co-visibility data	Mono-Camera	The feature used for loop closing and place recognition is the same feature used by the tracking and mapping tasks.	• Operate in a large environment in real-time. • It can perform a real-time loop closing via pose graph optimization. Ability to robustly do the initialization process.	It does not apply to Stereo and RGB-D cameras
Artal, et al. 2017	Bundle adjustment, place recognition, loop closing, and utilizing the co-visibility data	Monocular, stereo, and RGB-D cameras	The pre-processing feature-based allowed the system to apply to different types of cameras.	• Efficient for indoor and outdoor environments. • Low computation cost, which makes it a fast algorithm. Zero drift and robust localization with a lightweight model.	Suffer from motion blur issue.
Campos, et al., 2021	Relay on bundle adjustment and place recognition technique	Monocular, stereo, and RGB-D cameras	Precise IMU initialization scheme and multi-map merging. It's able to utilize the old information in all model phases	• Taking complete advantage of short-term, mid-term, and long-term data association that leads to zero drift. • Efficient for indoor and outdoor environments. Multi-mapping.	It fails to perform well in low-texture environments.
Pumarola, et al., 2017	Built over ORB-SLAM and based on Line Correspondences	RGB-D camera	Solve the problem of low texture	More accurate and efficient than ORB-SLAM.	The performance of PL-SLAM will be affected by the inter-frame rotation in the three images.
Wang, 2018	Using Convolutional Neural Network (CNN)	RGB-D camera	Decreases the gradient disappearance of the system and carry on in the training of deeper networks	Reducing the overall error compared to traditional CNN	System efficiency drops if the place or shape of objects changes.
Vincent, et al., 2020	Combining Extended Kalman filter and Deep Learning	RGB-D camera	Solve the issue of operating dynamic environments	• Fast operations. • It can be executed on a robot moving at a medium velocity. Operate in dynamic environments.	Unable to detect outliers segmentation which affects paradigm robustness.
Bruno and Colombini, 2021	Merging deep learning with a dynamic object probability model	Stereo camera and RGB-D camera	Present an efficient real-time vSLAM system for dynamic environments	• Has high accuracy and high stability in dynamic environments. Minimum drift and tracking error compared to ORB-	The detection scheme can't accurately estimate when the operating area is highly different from

				SLAM2.	the training area.
Bruno and Colombini, 2021	A hybrid system that utilizes deep learning with a modified version of geometry-based SLAM	Monocular camera	Introduced a system that can work efficiently in a highly dynamic environment.	• Robust without affecting the accuracy of the system. Work in both indoor and outdoor environments.	Not selecting the best features instead of using all extracted features.
Li, et al.,2021	Combining Deep Learning and ORB-SLAM2	Monocular and RGB-D cameras	Using multi-task feature extraction network with unsupervised feature points	• Can work in low texture environments without an efficient drop. The stability and accuracy of the system were improved	There is a hole compared to the conventional vSLAM from a real-time performance point of view
Maxime, et al., 2021	Key-point tracking via inverse Lucas-Kanade (LK) algorithm and pose estimation using the Huber coast model.	Monocular and stereo cameras	Present a system that is accurate, robust, and has a real-time operation	• Fill the gap of robustness, real-time operation, and accuracy. It can be applied in the ground and aerial robots for indoor and outdoor environments.	The algorithm deal with calibrated optical systems, can't perform calibration. It needs higher storage as it takes the whole image.
Chen, et al., 2021	Deep Learning, precisely Convolutional Neural Network (CNN).	Monocular camera	Presented a multi-constraint deep distance learning that can be used vSLAM	• Outperforms other traditional algorithms in terms of efficiency and effectiveness. Work in complex environments with dynamic changes.	Requires more geometric check or false-positive rejection ways.
Zhou, et al., 2021	Line and Points features operation with ORB-SLAM	Monocular camera	Solving the issue of operating in complex and weak textures environments with obvious brightness changes	• Robust and accurate operation. Decrease the drift and the loss.	Require longer processing time than ORB-SLAM
Liu and Miura, 2021	RDS-SLAM, Mask R-CNN, and dense optical flow	RGB-D camera	Overcome the slow speed of Mask R-CNN while preserving a high-performance operation.	• Effectively work in a dynamic environment. Keep the real-time operation with heavy segmentation ways.	Can't operate in an outdoor environment
Liu and Miura, 2021	Human detection, Unsupervised learning segmentation, and geometric constraints	RGB-D camera	Detecting and removing outliers of moving obstacles from the tracking process.	• More robust vSLAM. • Has a minor drift error. Archives good tracking performance	The initial value of K-means should be given manually.
Zhang, et al., 2021	Deep learning technique and geometry constraints of the traditional stereo VO	Stereo cameras	Improve the traditional VO, reduce the relative pose error and root-mean-square drift	• Increase positioning accuracy. Reduced the error drift	Didn't optimize the map points.

The presented methods here are a collection of the most successful and recent algorithms in the vSLAM field. Table (2) shows introduces a summary of these methods. The mentioned algorithms may be deployed to different mobile robots, including aircraft robots, land robots, and underwater mobile robots. Also, the table indicates that different camera types were utilized in vSLAM technology, such as monocular, stereo, and RGB-D cameras. It's worth mentioning that the current researches orientation is toward the AI-based vSLAM as it can provide more robust and less computation complexity algorithms (Duan, et al., 2010).

6. CONCLUSION

In the last decades, autonomous robots have obtained significant attention in both military and civil sectors, this lead to a kind of standardization of algorithms that are used for self-driving robots. An essential task for autonomous robots is locating themselves in the operating environment and building up the map for the surrounding area. Simultaneous Localization and Mapping (SLAM) is the leading key solution to this problem. This paper presents a detailed review of the state-of-the-art SLAM algorithms for the visual-based robots known as vSLAM Algorithms. Three different categories of vSLAM have been shown in this review study; Kalman filter-based, Graph-based, and AI-based vSLAM. In this scientific review the recent ideas and solutions for the vSLAM problem is shown. This comparative study shows the superiority of AI techniques coupled with RGB-D cameras over other systems in terms of accuracy and adaptability to different environments. Future works are intended to build an entirely autonomous system that can conduct security patrolling. The built system will be based on depth cameras (RGB-D) coupled with AI technique in conjunction with a state-space matrix that governs the priority of some landmarks during the patrolling tours.

7. References

1. Ai, Y.-b., Rui, T., Yang, X.-q., He, J.-l., Fu, L., Li, J.-b., & Lu, M. (2020). Visual SLAM in dynamic environments based on object detection. Defence Technology(2214-9147), 1-10. doi:10.1016/j.dt.2020.09.012
2. Alismail, H., Browning, B., & Lucey, S. (2016). Photometric Bundle Adjustment for Vision-Based SLAM. In Computer Vision – ACCV 2016 (pp. 324–341). Cham: Springer International Publishing. doi:10.1007/978-3-319-54190-7 20
3. Alsadik, B., & Karam, S. (2021). The Simultaneous Localization and Mapping (SLAM)-An Overview. SURVEYING AND GEOSPATIAL ENGINEERING JOURNAL, 2(1), 1-12.
4. Artal, M. R., & Tardos, J. D. (2017). ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. IEEE Transactions on Robotics, 33(5), 1255-1262. doi:10.1109/TRO.2017.2705103
5. Artal, R. M., Montiel, J. M., & Tardos, J. D. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics, 31(5), 1147-1163. doi:10.1109/TRO.2015.2463671
6. Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE TRANSACTIONS ON SIGNAL PROCESSING, 50(2), 174-188.
7. Benavidez, P., Kumar, M., & Jamshidi, M. (2014). Improving Visual SLAM Algorithms for use in Realtime Robotic Applications. 2014 World Automation Congress (WAC) (pp. 1-6). Waikoloa: IEEE. doi:10.1109/WAC.2014.7084333
8. Bensalem, S., Gallien, M., Ingrand, F., Kahloul, I., & Nguyen, T.-H. (2008). Toward a More Dependable Software Architecture for Autonomous Robots. IEEE ROBOTICS AND AUTOMATION MAGAZINE.
9. Birk, A., & Pfingsthorn, M. (2016). Simultaneous Localization And Mapping (SLAM). Encyclopedia of Electrical and Electronics Engineering: Wiley, 1-41.
10. Bruno, H. M., & Colombini, E. L. (2021). LIFT-SLAM: a deep-learning feature-based monocular visual SLAM method. Neurocomputing, 455, 97-110. doi:10.1016/j.neucom.2021.05.027
11. Campos, C., Elvira, R., Rodrıguez, J. G., Montiel, J. M., & Tardos, J. D. (2021). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. IEEE Transactions on Robotics, 1-17. doi:10.1109/TRO.2021.3075644
12. Chen, L., Jin, S., & Xia, Z. (2021). Towards a Robust Visual Place Recognition in Large-Scale vSLAM Scenarios Based on a Deep Distance Learning. Sensors, 21(1), 1-19. doi:10.3390/s21010310
13. Correll, N. (2016). Introduction to Autonomous Robots. Colorado: Wiely.
14. Davis, T. A. (2006). Direct Methods for Sparse Linear Systems. SIAM,. In SIAM Book Series on the Fundamentals of Algorithms. Florida.
15. Davison, A. J. (2003). Real-Time Simultaneous Localisation and Mapping with a Single Camera. Ninth IEEE International Conference on Computer Vision, (pp. 1403–1410).
16. Dellaert, F., & Kaess, M. (2006). Square root SAM: Simultaneous location and mapping via square root information smoothing. Journal of Robotics, 1, 1-32.
17. Dib, A., Beaufort, N., & Charpillet, F. (2014). A Real

Time Visual SLAM For RGB-D Cameras Based on Chamfer Distance and Occupancy Grid. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). Besançon.

18. Duan, C., Junginger, S., Huang, J., Jin, K., & Thurow, K. (2019). Deep Learning for Visual SLAM in Transportation Robots: A review. Transportation Safety and Environment, 1(3), 177–184. doi:10.1093/tse/tdz019
19. Durrant-Whyte, H., Rye, D., & Nebot, E. (1995). Localization of Autonomous Guided Vehicles. Munich, German.
20. Grisetti, G., K'ummerle, R., Stachniss, C., & Burgard, W. (2011). A Tutorial on Graph-Based SLAM. IEEE INTELLIGENT TRANSPORTATION SYSTEMS MAGAZIN, 1, 31-43. doi:10.1109/MITS.2010.939925
21. Inc, M. (n.d.). Monocular Visual Simultaneous Localization and Mapping. (MathWorks) Retrieved 8 18, 2021, from https://www.mathworks.com/help/vision/ug/monocular-visual-simultaneous-localization-and-mapping.html
22. Jajulwar, K. K., & Deshmukh, A. Y. (2013). Design of Mobile Robot Navigation System using vSLAM and Distributed Filter Techniques. IEEE Sixth International Conference on Emerging Trends in Engineering and Technolog. Nagpur. doi:10.1109/ICETET.2013.35
23. Jiménez, J. E., Devy, M., & Gordillo, J. (2016). Visual EKF-SLAM from Heterogeneous Landmarks. Sensors,, 16(4), 1-26. doi:10.3390
24. Julier, S. J., & Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building. IEEE International Conference on Robotics 8 Automation. Seoul.
25. Karam, S., Lehtola, V., & Vosselman, G. (2019). INTEGRATING A LOW COST MEMS IMU INTO A LASER BASED SLAM for INDOOR MOBILE MAPPING. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2, 149-156.
26. Khairuddin, A. R., Talib, M. S., & Haron, H. (2015). Review on Simultaneous Localization and Mapping (SLAM). IEEE International Conference on Control System, Computing and Engineering,, (pp. 85-90). Penang, Malaysia.
27. Leonard, J. J., & Durrant-Whyte, H. F. (1991). Mobile Robot Localization by Tracking Geometric. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 7(3), 376-382.
28. Li, G., Yu, L., & Fei, S. (2021). A deep-learning real-time visual SLAM system based on multi-task feature extraction network and self-supervised feature points. Measurement, 168, 1-10. doi:10.1016/j.measurement.2020.108403
29. Liu, Y., & Miura, J. (2021). KMOP-vSLAM Dynamic Visual SLAM for RGB-D Cameras using K-means and OpenPose. IEEE/SICE International Symposium on System Integration, (pp. 415 - 420). Fukushima. doi:10.1109/IEEECONF49454.2021.9382724
30. LIU, Y., & MIURA, J. (2021). RDMO-SLAM Real-Time Visual SLAM for Dynamic Environments Using Semantic Label Prediction With Optical Flow. IEEE Access, 9, 106981 - 106997.
31. Lu, F., & MILIOS, E. (1997). Globally Consistent Range Scan Alignment for Environment Mapping. Autonomous Robots , 4, 333–349.
32. Makhubela, J. K., Zuva, T., & Agunbiade, O. Y. (2019). Vision based Simultaneous Localization and Mapping in a light intensity Static Environment. International Journal of Scientific Research & Engineering Technology (IJSET), 13, 22-27.
33. Maxime, F., Alexandre, E., Juli, M., Martial, S., & Guy, L. B. (2021). OV2SLAM: A Fully Online and Versatile Visual SLAM forvReal-Time Applications. IEEE Robotics and Automation Letters, 6(2), 1399-1406.
34. Maybeck, S. P. (1979). Stochastic Models, Estimaton and Control, (Vol. 1). Ohio: Academic Press,.
35. MOHAMED, S. A., HAGHBAYAN, M. H., WESTERLUND, T., HEIKKONEN, J., TENHUNEN, H., & PLOSILA, J. (2019). A Survey on Odometry for Autonomous Navigation Systems. IEEE Access, 97466-97486.
36. Muhammed, N., Fosi, D., & Ainouz, S. (2009). Current state of the art of vision based SLAM. The International Society for Optical Engineering, (pp. 1-13).
37. Mur-Artal, R., Montiel, J., & Tard´os, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. IEEE TRANSACTIONS ON ROBOTICS, 1-17. doi:10.1109/TRO.2015.2463671

38. N. G. Hockstein, C. G. Gourin, R. A. Faust, & D. J. Terris. (2007, March 17). A history of robots: from science Wction to surgical robotics. Springer, 1, 113–118. doi:DOI 10.1007/s11701-007-0021-2
39. Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). DTAM: Dense Tracking and Mapping in Real-Time. IEEE International Conference on Computer Vision, (pp. 2320–2327).
40. Olson, E., Leonard, J., & Teller, S. (2006). Fast iterative optimization of pose graphs with poor initial estimates. The IEEE Int. Conf. on Robotics & Automation (ICRA), 2262–2269.
41. Pumarola, A., Vakhitov, A., & Agudo, A. (2017). PL-SLAM: Real-Time Monocular Visual SLAM with Points and Lines. IEEE International Conference on Robotics and Automation (ICRA). Singapore. doi:10.1109/ICRA.2017.7989522
42. Saeedi, S., Trentini, M., Seto, M., & Li, H. (2016). Multiple-Robot Simultaneous Localization and Mapping: A Review. Journal of Field Robotics, 33(1), 3-46. doi:0.1002/rob.21620
43. Siegwart, R., & Nourbakhsh, I. R. (2004). Intoduction To Autonomous Mobile Robot. London: The MIT Press.
44. Smith, R. C., & Cheeseman, P. (1986). On the Representation and Estimation of Spatial Uncertainty. The International Journal of Robotics Research, 5(4), 56-68.
45. Sorenson, H. W. (1966). Kalman Filtering Techniques. Advances in Control Systems, 3, 219-292.
46. Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: a survey from 2010 to 2016. Transactions on Computer Vision and Applications, 1, 1-11. doi:10.1186/s41074-017-0027-2
47. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press.
48. Vincent, J., Labbe, M., Lauz, J. S., Grondin, F., Comtois-Rivet, P.-M., & Michaud, F. (2020). Dynamic Object Tracking and Masking for Visual SLAM. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4974-4979). Las Vegas: IEEE.
49. Wallén, J. (2008). The history of the industrial robot. Linköpings universitet, Electrical Engineering. Linköping, Sweden: Linköpings universitet poblications.
50. Wang, X. (2018). Autonomous Mobile Robot Visual SLAM Based on Improved CNN Method. IOP Conference Series: Materials Science and Engineering, 466, pp. 1-8. Nanjing. doi:10.1088/1757-899X/466/1/012114
51. Whyte, H. D., & Bailey, T. (2006). Simultaneous Localisation and Mapping (SLAM) Part I The Essential Algorithms. IEEE Robotics & Automation Magazine, 1, 99-108.
52. Zhang, S., Lu, S., He, R., & Bao, Z. (2021). Stereo Visual Odometry Pose Correction through Unsupervised Deep Learning. Sensors, 21, 4735 - 4,753. doi:10.3390/s21144735
53. Zhou, F., Zhang, L., Deng, C., & Fan, X. (2021). Improved Point-Line Feature Based Visual SLAM Method for Complex Environments. sensors, 21(13), 1-17. doi:10.3390/s21134604