Expert Systems With Applications 205 (2022) 117734

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

A survey of state-of-the-art on visual SLAM

Iman Abaspur Kazerouni [a,b,*], Luke Fitzgerald [b], Gerard Dooly [b], Daniel Toal [b]

[a] CONFIRM Centre for SMART Manufacturing, University of Limerick, V94 C928 Limerick, Ireland
[b] Department of Electronics and Computer Engineering, Centre for Robotics and Intelligent Systems, University of Limerick, V94 T9PX Limerick, Ireland

ARTICLE INFO

Keywords:
SLAM
Feature matching
Sensors
Robot
Deep learning

ABSTRACT

This paper is an overview to Visual Simultaneous Localization and Mapping (V-SLAM). We discuss the basic definitions in the SLAM and vision system fields and provide a review of the state-of-the-art methods utilized for mobile robot's vision and SLAM. This paper covers topics from the basic SLAM methods, vision sensors, machine vision algorithms for feature extraction and matching, Deep Learning (DL) methods and datasets for Visual Odometry (VO) and Loop Closure (LC) in V-SLAM applications. Several feature extraction and matching algorithms are simulated to show a better vision of feature-based techniques.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is a wide and important topic in modern robotic and smart industry and can be used for both indoor and outdoor environments. Navigation, localization and mapping are basic technologies for smart autonomous mobile robots. SLAM plays an important role in smart manufacturing in either safe or high-risk and difficult to navigation industrial environments. Early research in SLAM was undertaken by Leonard and Durrant-Whyte in the early 1990s (Leonard & Durrant-Whyte, 1991) and the area has grown very fast, and many algorithms and techniques have been proposed to date.

SLAM can be defined as having two major parts: one, building a map of unknown indoor or outdoor environments and two, at the same time, track the position or movement of the sensors and camera (generally on a mobile robot) through different positions and different times in that environment.

SLAM can be used in a wide range of applications such as on air and underwater mobile robots, autonomous vehicles, drones, physical video games etc. Fig. 1 shows a SLAM technique flowchart for general algorithms.

Fig. 2 shows the formulation of SLAM to explain the problem. x_k is the robot state (orientation and position) at time k, u_k is the robot control input to move it from state x_{k-1} to x_k, z_k is the measurements by sensors and m_n is the landmark observed from the respective robot state. SLAM techniques are used to find the x and m from the u and z.

Table 1 shows the main SLAM surveys to date. This table covers the review literatures since 2015 so far to show the-state-of-the-art topics. In these reports, sensors used in SLAM systems and several types of SLAM are discussed and this paper is a comprehensive review from sensors to deep learning methods used in Visual SLAM.

This paper outline is as follows: Section 2 contains an overview of SLAM literature, then, in Section 3, sensors utilized in the SLAM approaches are discussed. Section 4 presents a review of feature extraction and matching algorithms with simulation results. Deep Learning (DL) methods and V-SLAM datasets are studied in a comparison view in Sections 5 and 6, respectively. Finally, a conclusion is drawn in Section 7.

2. SLAM

SLAM has been a popular topic of research in intelligent vehicles over the past three decades. The first principle research of this method was originally presented for autonomous control of mobile robots (Chatila & Laumond, 1985). Then, SLAM applications have been used in a wide range of topics such as augmented reality (AR) visualization, computer vision modeling, and self-driving cars(Taketomi, Uchiyama, & Ikeda, 2017). In recent years, SLAM has been used as a smart technique for building a 3D map of the environment using sensor fusion algorithms. Much recent research effort is focused on application in the automotive industry with increasing levels of auto/semi-auto car control.

Lu and Milios (1997) proposed a basic graph structured model for SLAM called Graph-SLAM to find the robot pose in an area based on the robot motion and observation data. As shown in Fig. 3, this method had two basic parts, Front-End and Back-End processes (Woo, 2019). In the

* Corresponding author at: CONFIRM Centre for SMART Manufacturing, University of Limerick, V94 C928 Limerick, Ireland.
E-mail addresses: iman.abaspur@ul.ie (I. Abaspur Kazerouni), luke.fitzgerald@ul.ie (L. Fitzgerald), gerard.dooly@ul.ie (G. Dooly), daniel.toal@ul.ie (D. Toal).

https://doi.org/10.1016/j.eswa.2022.117734
Received 25 February 2022; Received in revised form 12 May 2022; Accepted 31 May 2022
Available online 3 June 2022

Front-End process, the mobile robot pose is computed according to the output of sensors to produce a graph based on nodes and edges, while the Back-End process optimizes these nodes to the constraints of the edges. The loop closure system is used for further optimization and recognizing the previously visited places of the mobile robot. The loop closure detection gives the mobile robot the ability of recognizing the same scenes.

There are several methods to find the output parameters of SLAM and optimize the robot movements. Kalman and Bayes filters are two basic systems used for finding and estimating unknown parameters. The Kalman Filter assumes that Gaussian noises affect data. Kalman Filter is planned to solve the problems of linear systems in their basic form and are used for SLAM, although they have great convergence properties. In nonlinear filtering systems such as in SLAM, the Extended Kalman filter (EKF) is a common tool. EKF introduces a step of linearization for the nonlinear systems, and a first-order Taylor expansion performs linearization around the current estimate (Ullah, Su, Zhang, & Choi, 2020). The EKF is the core of the well-known method called EKF-SLAM. Leonard and Durrnat-Whyte (Leonard & Durrant-Whyte, 1991) used EKF to implement one of the basic SLAM technology systems. The mobile robot environment and motion are nonlinear systems, and the Kalman Filter cannot work properly on nonlinear systems and researchers and companies have been working on SLAM algorithms to adopt the basic ideas since the classical age (1986–2004). The Monte Carlo algorithm for localization is another filter to solve the nonlinear system problem which is the base of the Fast-SLAM (Montemerlo, Thrun, Koller, & Wegbreit, 2002) approach, but it is not so useful in a modern environment with large numbers of objects and images.

Montemerlo et al. (2002) proposed a model based on EKF algorithm and particle filter for non-Gaussian non-linear systems called Fast-SLAM. This method uses the separated conditional map and motion model parts to produce a smaller sampling space and reduce the dimensional state space. Currently, several filters are used to adopt SLAM techniques. For particle reduction, Rao-Blackwellised Particle Filter (RBPF) (Murangira, Musso, & Dahia, 2016; Zhao, Wang, Qin, & Zhang, 2018) and Kullback-Leibler Distance (KLD) sampling algorithm (Shiguang & Chengdong, 2017) methods are used in SLAM applications. RBPF produces a map using many particles with high computational cost to address the SLAM problem by determining the position at the first step and then reconstruct the map. This technique can be adopted using optimization algorithms to reduce process time and computation. At first, the posterior probability density is obtained by RBPF generated samples (particles) and then importance weights are calculated for each particle after observation. The particles are rearranged based on their weights that represent probability and importance and low weights samples can be eliminated in this stage and finally, the map is estimated based on the observation and trajectory with lower computational cost. After each observation, the particles weight will be updated for best map estimation. G-mapping is an open source SLAM based on particle filter, RBPF and scan matching algorithms (Grisetti, Stachniss, & Burgard, 2007). This technique reduces the samples and particles to choose effective particles and build a map with lower computation cost and time.

Milford, Wyeth, and Prasser (2004) presented a basic modern technique for SLAM with the path integration, visual association, and competitive attractor processes called Rat-SLAM. This model was used to recognize both unique and ambiguous landmarks. The visual association process is maintaining consistent representations of pose in the face of the inconsistent representations from the coarse path integration process and the competitive attractor dynamics is a block to ensure that the total activity in the pose cells is constant (Milford et al., 2004). A unique landmark is one that is uniquely distinguishable, ambiguous landmarks are those that are not quite as distinct, where one landmark can be confused for another because of their similarities.

A wide range of sensors such as camera, sonar, lidar, etc can be used for localization and mapping in robotic systems. Vision SLAM or V-SLAM refers to those SLAM systems which use cameras as the main input sensors to receive visual information of unknown objects and environments. Mono-SLAM is a V-SLAM technique for real time application which is developed by Davison (2003). This feature-based SLAM technique is the basis of modern SLAM for real time applications. Feature extraction is the key to this type of SLAM and there are several feature detection and extraction algorithms which can be used in Mono-SLAM. Klein and Murray (2007) presented separated tracking and mapping systems to reduce the computational cost of Mono-SLAM called Parallel Tracking and Mapping (PTAM). ORB-SLAM (Mur-Artal & Tardós, 2017; Mur-Artal, Montiel, & Tardos, 2015) is the adopted version of Mono-SLAM which works on extracted features using an ORB key-point feature descriptor algorithm. ORB-SLAM2 (Mur-Artal & Tardós, 2017), which is one of the most popular modern methods due to its robustness, builds upon the original ORB-SLAM approach, offering increased support for stereo and RGB-D camera setups.

ProSLAM (Schlegel, Colosi, & Grisetti, 2018) is an indirect/feature based V-SLAM system designed for ease of understanding and implementation. It builds local maps with landmarks acquired in a nearby region. These local maps are built through the process of ProSLAMs four core modules: triangulation, incremental motion estimation, map management and relocalization. This system is shown to have a competitive performance when compared to other well know SLAM systems while having a lower computational load, but the lack of optimizations such as Bundle Adjustment used in other approaches allows it to fall behind in terms of robustness to errors (Gao, Lang, & Ren, 2020).

The Iterative Closest Point (ICP) and scan matching algorithms are widely used for pose estimation and mapping in SLAM techniques (Donoso, Austin, & McAree, 2017; Grisetti et al., 2007). In these methods, the initially calculated mobile robot pose is iterated, and the scan information is matched and optimized to build an accurate map of the area. Fig. 4 shows the three scan matching methods for SLAM techniques (Xuexi, Guokun, Genping, Dongliang, & Shiliu, 2019).

Direct Methods for visual SLAM are techniques which use original images directly as input data instead of key-point features and reconstruct all points instead of only edges and corners. Dense Tracking and Mapping (DTAM) is a direct RGB video based method proposed by Newcombe, Lovegrove, and Davison (2011). Depth information and Camera motion are estimated in this algorithm to build a 3D model of an area. Ondrúška, Kohli, and Izadi (2015) proposed an adopted DTAM model for real time cell phone applications. Engel, Schöps, and Cremers (2014) presented a direct monocular SLAM called Large-Scale Direct Monocular (LSD-SLAM). This model requires depth estimates of each pixel and uses the image pixels in a dense map instead of feature extraction. Camera motion and pixels depth values are estimated in this algorithm and can be used for low quality and stereo images. Engel, Koltun, and Cremers (2017) presented a Visual Odometry (VO) method based on a Direct Sparse and Direct Structure and Motion system Direct Sparse Odometry (DSO). This method uses both photometric and geometric camera calibration results and divides an image into different blocks to select high intensity points as reconstruction samples. DSO uses both direct and sparse models as a monocular visual odometry algorithm to optimize the real time performance. VO is the estimating of the motion of an object in the camera image frames and camera motion

Fig. 1. General SLAM flowchart.

between adjacent images and is based on Visual SLAM techniques.

Liu et al. (2020) presented a tightly coupled EKF framework for Inertial Measurement Units (IMU) called Tight Learned Inertial Odometry (TLIO). This method uses deep learning algorithms to train a network which learns a priori on the displacement distributions from statistical motion patterns. They show that with prior learning, an IMU sensor can generate enough information for calibration and low drift pose estimation. Drift is the buildup of pose estimation inaccuracies, resulting in an increasingly displaced predicted position in comparison to the real-world pose, corrected at the point of loop closure. Less inaccuracies allows for simpler correction, and greater reliability for real-time systems. The neural network in this system is forced to learn only from prior information of typical human motion and does not use model-based state propagation. Karto-SLAM is a graph optimization model with only a point graph (robot pose) proposed by Konolige et al. (2010). This algorithm uses sparse pose adjustment and a non-iterative Cholesky matrix for a direct nonlinear optimization solution. This technique uses Spare Pose Adjustment (SPA) for matching and loop closure, which is a method that builds upon the Levenberg-Marquardt algorithm, making it efficient for sparse systems found in the building of 2D maps.

Kohlbrecher, Von Stryk, Meyer, and Klingauf (2011) proposed a robust scan matching technique based on a LiDAR system called Hector-SLAM. This algorithm has two main sections which include an integrated 3D navigation system and a 2D LiDAR- SLAM system without explicit loop closure detection. The scan matching technique estimates the rotation and translation of the mobile robot between two scans using nearest neighbor scan matching (Xuexi et al., 2019).

In recent years, RGB-D cameras are widely used for V-SLAM techniques in real time applications. Image processing, machine vision and deep learning algorithms are core of these systems to decrease computational cost and time and to increase the accuracy and ability of mobile robots for 3D mapping, localization, texture information, object detection and other industrial goals. A general RGB-D SLAM process can be arranged as:

- RGB-D camera motion estimation using Iterative Closest Point (ICP) (Chen & Medioni, 1992)
- Apply machine vision techniques to detect objects and eliminate moving objects
- 3D Reconstruction view of area using depth mapping

Zou and Tan (2012) proposed a camera pose estimation and mapping model for dynamic objects and environments in the SLAM process called Co-SLAM. This technique follows conventional sequential structure-from-motion methods and works on a sequence of tasks that include camera pose estimation, mapping, point classification and camera

Table 1
The SLAM review literatures.

Ref.	Discussed Topic	Year
(Chong et al., 2015)	SLAM Sensor Technologies	2015
(Yousif, Bab-Hadiashar, & Hoseinnezhad, 2015)	Visual Odometry and Visual SLAM for Mobile Robotics	2015
(Lowry et al., 2015)	Visual place recognition	2015
(Cadena et al., 2016)	Past, present, accomplishments and future challenges of SLAM	2016
(Saeedi, Trentini, Seto, & Li, 2016)	Multiple-robot SLAM systems	2016
(Taketomi et al., 2017)	Visual SLAM algorithms from 2010 to 2016	2017
(Zaffar et al., 2018)	Sensors used in SLAM	2018
(Jamiruddin, Sari, Shabbir, & Anwer, 2018)	RGB-D SLAM	2018
(Chen et al., 2018)	Visual SLAM	2018
(Zhao, He, Sani, & Yao, 2019)	Underwater SLAM	2019
(Duan, Junginger, Huang, Jin, & Thurow, 2019)	Deep Learning for Visual SLAM in Transportation Robotics	2019
(Yang et al., 2019)	LiDAR SLAM	2019
(Debeunne & Vivet, 2020)	Visual-LiDAR Fusion based SLAM	2020
(Kolhatkar & Wagle, 2022)	Indoor mobile robot LIDAR and RGB-D SLAM	2020
(Xia et al., 2020)	Semantic SLAM	2020

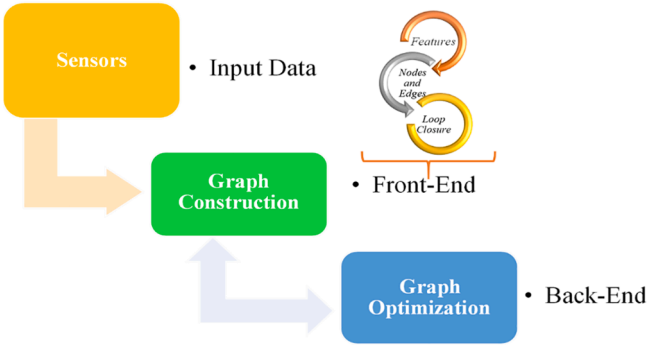
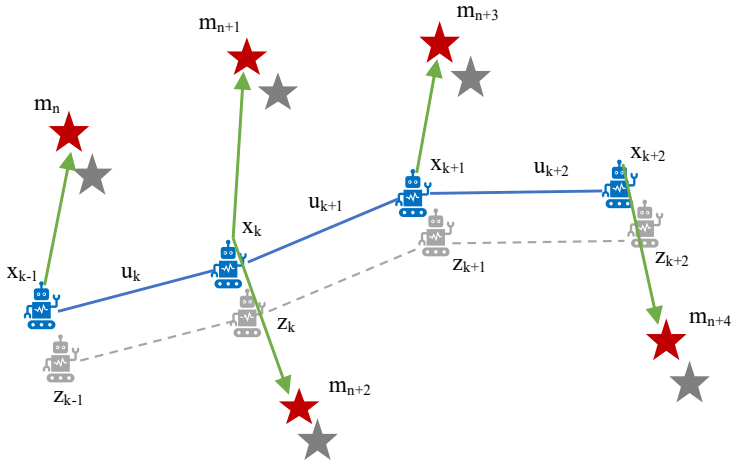


Fig. 3. Block diagram of Graph-SLAM.



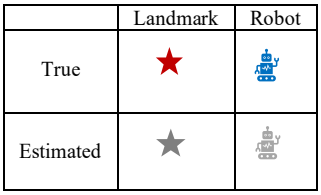


Fig. 2. The SLAM plan inputs and outputs.

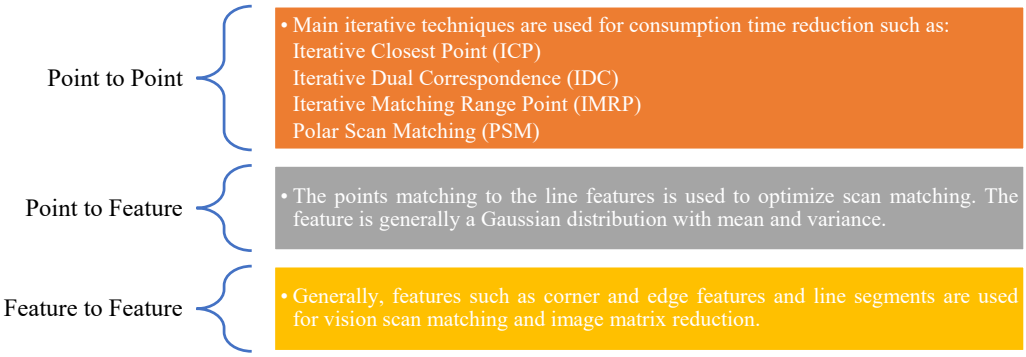


Fig. 4. The scan matching methods based on points and features.

grouping.

Kaess, Ranganathan, and Dellaert (2008) proposed a SLAM method based on fast incremental matrix factorization called incremental smoothing and mapping (i-SAM) which allows access to the underlying estimation uncertainties. Matrix factorization being a technique commonly used for filtering based recommendation, facilitating the generation of latent features, and allowing prediction to be made. However, this is not a perfect model to overcome vision systems problems. The algorithm incrementally maintains the square root information matrix by applying matrix factorization updates and the efficiency and accuracy of the new method need improvement which are updated in new version.

Milford and Wyeth (2012) presented a visual route-based navigation called SEQ-SLAM. This method is based on finding the local best matches and sequence recognition components to calculate the best sample matching location in every local navigation sequence.

Newcombe, Izadi, et al. (2011) presented a model for real time dense surface tracking and mapping using Kinect sensor equipment called Kinect Fusion. This technique is invariant to light changing and builds a 3D map by fusing depth maps into the voxel space with high accuracy and low processing time. Voxels represent a value on a grid in 3-dimensional space. Voxel space is a 3D grid environment.

Rossi et al. (2018) developed a technique similar to Kinect fusion for underwater application called Stereo Fusion. The infra-red projected light component of the Kinect sensor is not suitable for underwater application so the 3D geometry in stereo Fusion is derived through stereo imagery. And the technique uses GPU technology for image processing at camera frame rates.

Continuous-Time SLAM (CT-SLAM) is a continuous-time trajectory model for combining high-rate asynchronous sensors (Dubé, Sommer, Gawel, Bosse, & Siegwart, 2016; Park et al., 2018). This technique can improve the motion distortion problem in LiDAR sensors, which appears as inaccuracies in the plotted map, needing to be compensated for afterwards. Computational cost and operation time are high, and optimization is a vital aspect for use of this system.

SLAM++ is a real time 3D object-oriented method based on Kinect Fusion proposed by Salas-Moreno, Newcombe, Strasdat, Kelly, and Davison (2013). This method tracks and builds a map of 3D scenes at the object level in a real time system. In this model, ICP is used to track objects to produce the edge measurements for an object-level pose-graph system. The SLAM++ process can be defined as:

- Surface measurement calculation in the form of a regular map
- Camera pose tracking using ICP
- Object detection using provided database
- Update data using second ICP estimation
- Adding correct object into the map
- Build a full depth map and object detection by rendering objects from the SLAM graph

Steckel and Peremans (2013) proposed a functional model for mobile robotic sonar sensing called Bat-SLAM. This model is an updated Rat-SLAM using a biomimetic sonar sensor instead of a vision sensor and a Continuous Attractor Network (CAN) neural network is used for spatial orientation and map building. An attractor network is one that converges toward a stable pattern over time. There are different types of attractor network, each more useful depending on what is being modelled. The continuous type is good for continuous variables such as position in space.

Whelan, Leutenegger, Salas-Moreno, Glocker, and Davison (2015) proposed a real time dense visual SLAM for building a surfel based map using an RGB-D camera and called the technique Elastic Fusion SLAM. Surfels, or surface elements, are small discs that represents surface patches, and can be used as an alternative to a polygon. This technique is based on the Keller et al. (2013) method with an added loop closure system and provides a camera trajectory and dense surfel map. In Elastic Fusion, the vertex shader transforms surfel positions into the camera view and uses the normal vector and surfels radius to produce a square plane, while the fragment shader discards any fragment of the square which is outside of the circular radius to render the final surfel (McCormac, 2018). Vertex and fragment shaders are basically types of functions defined by the developer for use in graphics programming.

As described in this section, there are a wealth of V-SLAM methods to choose from. Their applicability largely depends on their intended use and the environment in which they are to be deployed. These methods are often categorized as either indirect or direct models (Chen, Zhou, Lv, & Deveerasetty, 2018). The indirect models being those that make use of feature extraction and matching, while the direct methods estimate motion based on pixel intensities. In terms of mapping models, what is commonly seen are metric representations which encode the geometry of the environment, but there has also been a desire to explore semantic based representations to extract meaning from the surroundings, which can allow for greater intelligence in autonomous navigation (Cadena et al., 2016), (Xia et al., 2020), (Radwan, Valada, & Burgard, 2018).

In previous comparison review literature (Gao et al., 2020), it was found that ORB-SLAM2 is a strong performer among indirect methods when it comes to translation and rotation Root Mean Square Errors (RMSEs), especially noted in faster motion applications, while Stereo DSO works well in comparison to other direct methods. In general, direct methods appeared to have better results in rotation estimation using illumination information, while indirect methods accel in translation estimation by tracking important features.

The main objective of SLAM problem is building a map of an environment using a mobile robot. This map has applications in robot navigation, manipulation, tele-presence, semantic mapping and unmanned vehicles and in planet rovers (Ajay & Venkataraman, 2013). One of the challenges with V-SLAM is solving the loop closure problem using visual data in real-time situations. The difficulty of this challenge is in the environment situation changes which can affect robot performance due to dynamic factors such as weather, illumination, noise, etc.

Towards the improvement of accuracy and robustness, a sensor fusion approach is seen to be desirable. While cameras can capture rich details, alternative data streams from sensors such as Sonar or LiDAR can complement this as discussed in the following section. The modern introduction of large-scale datasets has also driven the deep learning contribution to V-SLAM which is outlined later in this paper.

3. Sensors

The ability to sense or detect the real space and environment is one criterial element in SLAM and robot systems. Autonomous robots should have proper sensors to receive and process the best information and signals from the environment to build a map of the space. Currently SLAM applications often utilize cameras as sensors. Cameras can facilitate the users to adopt applications for more capability such as object detection, segmentation and other vision base systems. There are several types of sensors applied in SLAM such as RGB-D cameras, LiDAR, etc. Table 2 shows sensors commonly employed for SLAM.

Sound Navigation and Ranging (SONAR) sensors detect an object from the echo of an ultrasonic signal bounced off it and are widely utilized for mobile robots. SONAR sensors operation is based on sound signals and can work in dark environments and are common for underwater robots where the use of light-based imaging faces challenges. Sonar can cover a wide space for mapping and navigation, but they are not suitable for object detection goals because they cannot detect objects corners with high accuracy. There are a wide range of commercial acoustic sensors with different power consumption and operation frequency.

A SLAM solution based on a 2D laser scanner is a common SLAM model reported in the literature. These approaches are generally applicable where the motion of the robot is constrained e.g., a wheeled robot moving on a plane or a flying robot flying at a constant altitude or constant height above ground trajectory. Using laser sensors, robots can build a map of indoor and outdoor spaces. Laser sensors with 20 Hz update rate can often be utilized for indoor SLAM robots. Like acoustic sensors, lasers are not a proper choice for object detection techniques, but they can be used for 2D and 3D mapping with high speed processing called tiny SLAM (Steux & El Hamzaoui, 2010). It is easier to apply to scenarios where height/altitude for example is constant. If this value varied, it would add a degree of complexity in using the 2D laser where it is necessary to identify the change using another sensor and fuse this information together to compensate for mapping issues.

LiDAR based SLAM is one of the modern mapping systems broadly applied in robotic systems. Although previous versions of LiDAR are usually bulky and heavy sensors, recently small and light LiDARs are available for applying in fast and precise imaging processes. LiDAR is analogous to RADAR but uses light from a laser instead of radio waves to detect/image an area, objects and depth up to 300 m. LiDAR can build a precise map of indoor and outdoor areas (Hess, Kohler, Rapp, & Andor, 2016).

In recent years, object detection methods have been added to robotic systems to design a comprehensive map of an area with capability to distinguish recognizable objects within the environment. RGB-D cameras are a common choice for this purpose to find objects and detect corners and contours for clustering. RGB-D cameras include IR transmitters, IR receivers and monocular camera and make an RGB image with structured light patterns in infrared transmitted on an area and received, giving depth (or range) of each image pixel (Sturm, Engelhard, Endres, Burgard, & Cremers, 2012).

Monocular Cameras are common, cheap and generally readily available sensors included as standard on many robots, and these can be used for SLAM techniques. Thousands of algorithms, source code, books and technical papers are available for image processing, machine vision and deep learning based on camera outputs which give RGB images. There are several algorithms to build a map of an environment and for object detection using data fusion and image analysis. Monocular cameras are low power sensors and are suitable for low budget SLAM projects (Civera, Davison, & Montiel, 2008; Engel et al., 2014; Mur-Artal et al., 2015).

Stereo cameras are another vision system for SLAM which can estimate depth in images using two captured pictures from an object with two different angles. The RGB-D sensors has led to great progress in SLAM (Dryanovski, Valenti, & Xiao, 2013). The advantages of these systems lie in the low cost and high mobility. However, RGB-D sensors have some significant drawbacks in dense 3D mapping systems. These sensors can be used for a limited distance and a limited field of view (FoV). This can cause tracking loss due to lack of the spatial structure needed to constrain ICP (iterative closest point) alignments (Tang et al., 2016).

Fig. 5 shows the stereo vision geometry for stereo camera operation model.

From Fig. 5, the depth of the object point (Z) can be obtained as:

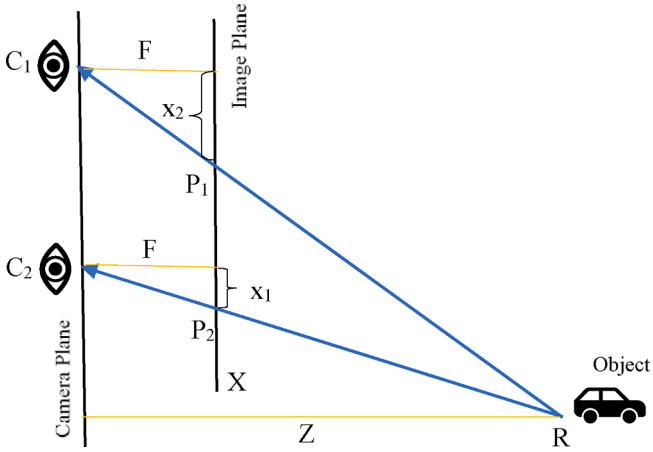


Fig. 5. The stereo vision geometry.

Table 2
SLAM sensors.

Sensor	Application	Range (Frame Rate for cameras and Operating Frequency for acoustics)	Price (€)	SLAM Type	Power Consumption (W)
Sonar	Underwater	1 kHz and 500 kHz	500	Bat SLAM	0.01–5
Laser	Indoor, Outdoor and Drones	1–100 Hz	200	Laser-based SLAM, TinySLAM	1–100
LiDAR	Indoor, Outdoor and Drones	1–50 Hz	100–5 K	LiDAR SLAM, CT-SLAM	5–200
RGB-D	Indoor	10–400 Hz	150–500	RGB-D SLAM	0.3–5
Monocular Camera	Indoor and Outdoor	20–200 Hz	100–5 K	Mono SLAM	0.01–10
Stereo	Indoor, Outdoor and UAV	1–400 Hz	400–4 K	Stereo LSD-SLAM	2–15

$$\begin{cases} \dfrac{X}{Z} = \dfrac{x_2}{F} \\ \dfrac{X - C_1 - C_2}{Z} = \dfrac{x_1}{F} \\ Z = \dfrac{F(C_1 - C_2)}{x_2 - x_1}, \end{cases} \tag{1}$$

where F is focal length of the cameras, C_1 and C_2 are Camera1 and Camera2 positions and x_1 and x_2 are image space coordinates. A 3D view of an area can be obtained by knowing depth of each point in stereo imaging. (x_2 - x_1) is the disparity defined as the distance between points in image plane corresponding to camera center and the scene point 3D. Fig. 6 shows the disparity map of two different angle images. Bright white pixels mean that the object is near the cameras and depth for each object can be estimated based on the disparity map.

Solving SLAM problems is dependent on the sensing technologies and price, power consumption, real time processing, size and noise removal ability are the main factors for autonomous robots. Acoustic sensors generally cannot detect and extract the main features of an unknown area at sufficient resolution. LIDARs tend to be bulky expensive sensors and have pose estimation problems. Monocular cameras lack depth information and RGB-D cameras can obtain depth information and objects features but they have limited range (Zaffar, Ehsan, Stolkin, & Maier, 2018).

4. Feature extraction & matching

Camera-based SLAM detects the visual features of the environment which human vision systems can automatically distinguish such as corners, edges, colors, shadows, and depths. Feature extraction is the most important part of image matching for design/building a map of an area and for object detection techniques. In recent years, use of high-quality, high-resolution cameras, big data images with large details are generated as camera output (Schlegel et al., 2018), (Xuexi et al., 2019). Processing of large data can be slow and results in the need for fast and high-capacity processors which often are expensive for research labs and target robot systems. These processors consume a large amount of power to run and are not ideal for mobile robots. Feature extraction can decrease the size of input data using selected data and features of input images. There are several methods and algorithms reported for feature extraction in the literature which can be used and applied for SLAM. Features in an image are pixels with common properties and distinctive from other proximal pixels. Features for object detection techniques can be image details such as texture, mean of pixels, color, corner, etc. and for vision SLAM must be invariant to rotation, orientation, translation, scaling and luminous intensity. For mobile robots, the vision system can be very sensitive, and it is important to be able to find the pose of surrounding objects and find a collision free path. Each frame of the video stream received from cameras should be compared with previous frames and this real time system needs a fast algorithm to find and extract the best image features and remove other data. It is important to have robust methods for identifying current robot position so it can avoid obstacles in the area and continue mapping accurately. If everything is done properly, the path it takes will be collision free, and the system will have a complete map showing the boundaries of the area and any obstacles it maneuvered around.

The Gabor wavelets technique (Lee, 1996) is one of the basic image processing and machine vision algorithms for feature extraction work based on texture description using image decomposition into different orientations and scales. Gabor filters often are useful for images with different texture and segments (Zhang, Wong, Indrawan, & Lu, 2000). This basic technique also has been used in medical robots for texture classification along with fuzzy techniques (Singh & Singla, 2020). For indoor SLAM and object detection, the Gabor filter can be helpful because for example in an industrial environment, objects have strong lines, contours and edges at several different scales and orientation and the statistics of these features are very appropriate for object detection (Dewi, Sundararajan, Prabuwono, & Cheng, 2019).The general function G(x,y) of a two dimensional image Gabor filter can be defined as (Sun, Bebis, & Miller, 2006):

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} exp[2\pi jW\widetilde{x}] exp\left[-0.5\left(\left(\frac{\widetilde{x}}{\sigma_x}\right)^2 + \left(\frac{\widetilde{y}}{\sigma_y}\right)^2 \right) \right]$$

$$\widetilde{x} = x\cos\theta + y\sin\theta \ , \quad \widetilde{y} = y\cos\theta - x\sin\theta, \tag{2}$$

where σ is the scaling parameters of the Gabor filter for the x and y dimensions, W is center frequency and θ is filter orientation. This filter is like a local bandpass filter. This filter has been shown to possess optimal localization properties in both frequency and spatial domain and is suitable for texture segmentation problems. Gabor filter allows a certain band of frequencies and rejects the others. A Gabor filter can be shown as a sinusoidal signal of frequency and orientation, modulated by a Gaussian wave.

Generally, wavelets are multiresolution function methods for image decomposition. These algorithms output compact valuable data from images in multi-level resolution which have edge data for objects in each image. There are several fast wavelet algorithms like Haar wavelets used in object detection literature. The Discrete Wavelet Transform (DWT) is another feature extraction technique based on multiple wavelets (Gokulalakshmi, Karthik, Karthikeyan, & Kavitha, 2020). These techniques can be used for high frequency sub band images with different spatial orientations and frequencies.

In SLAM, to describe the different angles, orientation, and patterns of a specific object within one single image, the texture attribute can be useful, as texture analysis methods supply objective conditions for carrying out the object recognition and classification (Xu, Zhang, Li, Liu, & Zhu, 2021). The high dimensionality of a feature vector in a high-quality

(a) (b) (c)

Fig. 6. Stereo camera image for depth estimation (a) and (b) two original images with different view (Scharstein & Pal, 2007) and (c) shows the disparity map for stereo camera.

image that represents texture attributes limits its computational efficiency. Thus, it is inevitable to apply a method that combines the representation of the texture with a decrease in dimensionality, in a way to make the object detection algorithm more effective, faster, and computationally treatable. Principle Component Analysis (PCA) (Pearson, 1901) is another feature detection technique applied widely for many object-based analysis applications for data reduction. PCA transforms multi-dimensional data to a linear vector for dimensional reduction using determination of linear variables with largest variance. Currently, adopted versions of PCA are used for better feature extraction and image matrix dimension reduction.

An adopted PCA technique is 2-directional 2-dimensional principal component analysis ((2D)^{2}PCA). In this version, a 2-dimensional PCA is applied on the row direction of images, and then another 2-dimensional PCA is applied on the column direction of images. In this method, the size reduction is applied in the rows and columns of images simultaneously (Kazerouni & Haddadnia, 2014). 2-directional 2-dimensional principal component analysis is an accurate and fast data representation, feature extraction and image matrix dimensional reduction technique that aims at finding a more compact and less redundant representation of image data in which an effective reduced number of components can be independently responsible for image data variation (Abaspur Kazerouni, Dooly, & Toal, 2020). Given a 2-dimensional PCA operator in an image A with m rows and n columns we can define the covariance matrix C as:

$$C = \frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{m}\left(A_k^{(i)} - \overline{A}^{(i)}\right)\left(A_k^{(i)} - \overline{A}^{(i)}\right)^T \mathrm{AA} \tag{3}$$

where M is the training sample with m by n matrices, which are shown by A_k (k = 1, 2,…, M) and $\overline{A}$ and C represent the average matrix and covariance matrix respectively and $A_k^{(i)}\mathrm{A}_k^{(i)}$ and $\overline{A}^{(i)}$ $\overline{\mathrm{A}}^{(i)}$ denote the i-th row vectors of A_k and $\overline{A}$ $\overline{\mathrm{A}}$ respectively. Another 2-dimensional PCA can be applied in the image columns as:

$$C = \frac{1}{M}\sum_{k=1}^{M}\sum_{j=1}^{n}\left(A_k^{(j)} - \overline{A}^{(j)}\right)\left(A_k^{(j)} - \overline{A}^{(j)}\right)^T \mathrm{AA} \tag{4}$$

where $A_k^{(j)}\mathrm{A}_k^{(i)}$ and $\overline{A}^{(j)}$ $\overline{\mathrm{A}}^{(i)}$ denote the j-th column vectors of A_k and $\overline{A}$ $\overline{\mathrm{A}}$ respectively, and q first high eigenvalues of matrix C are located as columns in the matrix Z which $Z \in R^{m\times q}\mathrm{Z}\in\mathrm{R}^{\mathrm{m}\times\mathrm{q}}$. Projecting the random matrix A onto Z yields a 'q by n' matrix $Y = Z^TAY = \mathrm{Z}^{\mathrm{T}}\mathrm{A}$ and projecting the matrix A onto Z and X generates a 'q by d' matrix $= Z^TAX$ $\mathrm{Y} = \mathrm{Z}^{\mathrm{T}}\mathrm{AX}$.

Linear discriminant Analysis (LDA) (Balakrishnama & Ganapathiraju, 1998) is another feature extraction and data reduction technique which projects a high-dimensional feature vector to a low-dimensional space. This technique is usually used along with a PCA algorithm for image matrix reduction.

Finding the best features in images which are orientation, angle and contrast invariant is an important key for achieving higher accuracy and faster results in navigation and SLAM. Currently, several modern techniques have been proposed in the literature for key-point feature extraction and matching. The best examples include the following: Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Features from Accelerated Segment Test (FAST), Binary Robust Independent Elementary Features (BRIEF), Oriented FAST and Rotated BRIEF (ORB), Learned Invariant Feature Transform (LIFT), Fast Retina Key-point (FREAK), Binary Robust Invariant Scalable Key-points (BRISK), Histogram of Oriented Gradients (HOG), A Fast-Local Descriptor for Dense Matching (DAISY), Gradient Location and Orientation Histogram (GLOH), Local Intensity Order Pattern (LIOP), MatchNet, MROGH, KAZE and A_KAZE. When a smart system checks the features of sequence frames from a video stream in a real time system, feature matching algorithms find the common features in those frames and based on the orientation and depths of images, estimate the position and movement of the camera. To identify similar features, feature matching descriptors are compared across the images. Assuming the selected technique is effective, these feature descriptors will be unique, making them identifiable in subsequent frames of a video stream.

Dalal and Triggs (2005) presented a basic model of Histogram of Oriented Gradients (HOG) for human detection. In this feature descriptor algorithm, the main features are the distribution (histograms) of directions of gradients (oriented gradients). Using image gradients, the objects' corners and edges can be extracted and defined as main object features. This algorithm is one of the basic algorithms before introducing deep learning techniques for feature descriptors. Fig. 7 shows the magnitude of gradient for an image and HOG result.

Lowe (2004) proposed an image rotation and scale invariant algorithm called SIFT. This technique is widely used for key-point feature matching and is very robust in image scaling and rotation, while being partially invariant to illumination changes and affine transformations. This algorithm, for accurate key-point localization, uses the Taylor expansion of the Difference-of-Gaussian (DOG) scale-space function, D (x, y, σ), shifted so that the origin is at the candidate point (Lowe, 2004):

$$D(\mathrm{x}) = D + \frac{\partial D^T}{\partial \mathrm{x}}\mathrm{x} + 0.5\mathrm{x}^T\frac{\partial^2 D}{\partial \mathrm{x}^2}\mathrm{x}, \tag{5}$$

where D and its derivatives are evaluated at the candidate point, x= (x, y, σ) $^{\mathrm{T}}$ is the offset from this point and the location of the extremum is defined as:

$$\hat{\mathrm{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathrm{x}^2}\frac{\partial D}{\partial \mathrm{x}} \tag{6}$$

And for rejecting unstable extrema with low contrast, the function value at the extremum is expressed by:

$$D(\hat{\mathrm{x}}) = D + 0.5\frac{\partial D^T}{\partial \mathrm{x}}\hat{\mathrm{x}} \tag{7}$$

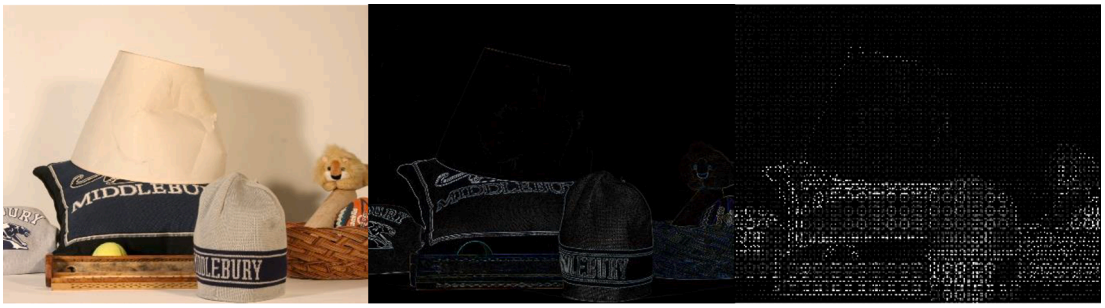
SIFT is useful for feature matching and image mosaicking (Rublee, Rabaud, Konolige, & Bradski, 2011), however, its large number of calculations makes it difficult to use in real time systems. It performs better in blob rich regions rather than corner rich regions (see Fig. 8(b)). Fig. 8 shows the SIFT method key-point extraction and image matching result.

To use a SLAM approach for mobile robots which often have limited computational resources, a useful algorithm for solving and dealing with SIFT difficulty is FAST which is a corner detection algorithm proposed by Rosten and Drummond (2006). For each pixel in the image, FAST considers neighboring pixels within a specific radius. This algorithm compares each pixel with the contiguous pixels in the circle to check whether they are brighter or darker than that pixel with a threshold. If a set of specific contiguous pixels in the circle are all brighter than the intensity of the candidate pixel plus a threshold value or all darker than the intensity of candidate pixel p minus a threshold value, then the candidate pixel is a corner pixel. The FAST corner detector is very fast, but for noisy images cannot extract correct features and selection of a proper threshold level can be difficult for some images.

SURF is a faster version of SIFT proposed by Bay, Ess, Tuytelaars, and Van Gool (2008). It generates multi-levels of image and descriptor pairs. The SURF algorithm works like SIFT but uses the Haar wavelet instead of using Difference-of-Gaussians. The SURF algorithm uses the Hessian matrix determinant for selecting the location and the scale to generate a fast and accurate descriptor. An approximation for the determinant of the Hessian can be defined as:

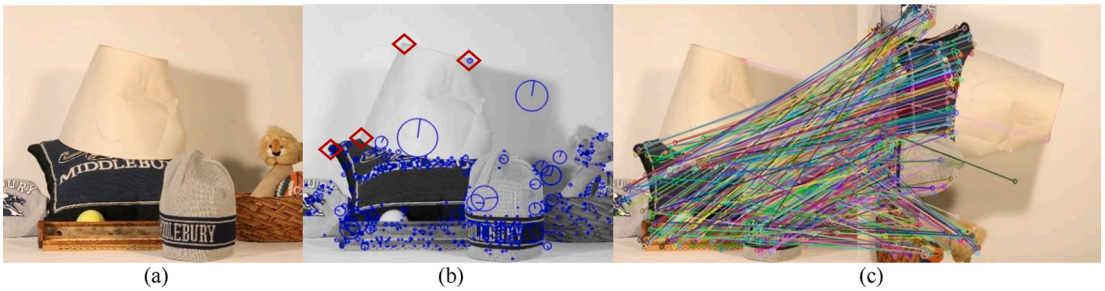
$$\det\left(H_{approx}\right) = D_{xx}D_{yy} - 0.81D_{xy}^2 \tag{8}$$

where D_{xx}, D_{yy} and D_{xy} are approximations for Gaussian second order derivatives for highest spatial resolution. Computational cost and calculations in the SURF method are more feasible and suitable for real time application but SIFT shows more robustness with important features and details. Fig. 9 shows the SURF algorithm feature extraction and



(a) (b) (c)

Fig. 7. HOG feature detection algorithm (a) original image (b) the magnitude of gradient and (c) HOG result.



(a) (b) (c)

Fig. 8. SIFT algorithm (a) Original image, (b) key-point detection (blobs: blue circle, corner: red diamond) and (c) Image matching for original image and 90 degrees rotated original image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

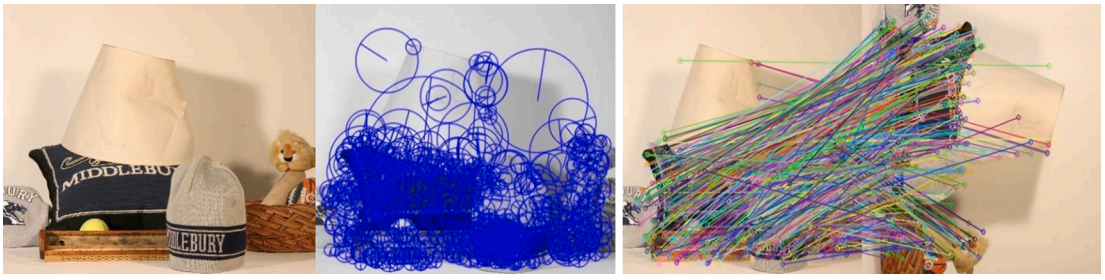


Fig. 9. SURF algorithm (a) Original image, (b) key-point detection (gray-scale level) and (c) Image matching for original image and 90 degrees rotated original image.

image matching result.

Calonder et al. (2011) presented a descriptor algorithm with more efficiency in real time applications called BRIEF. Binary Robust Independent Elementary Features (BRIEF) is a very noise sensitive algorithm and uses Gaussian kernel to smooth the image and then convert the smoothed image to a binary feature vector. The BRIEF feature descriptor can be defined as:

$$f_n(p) = \sum_{1 \le i \le n} 2^{i-1} \tau(p; x_i, y_i), \tag{9}$$

where p is image patch (the neighborhood around pixel), τ is binary test

(a) (b)

Fig. 10. BRIEF algorithm (a) Original image and (b) Image matching for original image and 90 degrees rotated original image.

for the selected set of n(x, y). Choosing the proper test points is vital for this algorithm. Fig. 10 shows a BRIEF method image matching result for a sample image.

The ORB (Oriented FAST and Rotated BRIEF) algorithm is another descriptor proposed by Rublee et al. (2011). This method works based on the BRIEF descriptor and FAST key-point detector and measures corner orientation by using an intensity centroid. The moments of an image patch can be expressed as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{10}$$

With these moments, the center of mass of the patch can be defined as:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{11}$$

Using C, the vector from the corner's center, O is computed as OC. The orientation of the patch is defined as:

$$\theta = atan2(m_{01}, m_{10}) \tag{12}$$

Using θ, it is possible to rotate it to a canonical rotation to compute the descriptor, thus obtaining some rotation invariance. The ORB algorithm is very fast, but it is less effective in terms of scale. This method is used in the SLAM approaches called ORB-SLAM (Mur-Artal et al., 2015) and ORB-SLAM2 (Mur-Artal & Tardós, 2017) which are feature-based SLAM techniques which generate keyframe based on a graph using ORB feature descriptor. Fig. 11 shows the ORB image matching result for a sample image.

Yi, Trulls, Lepetit, and Fua (2016) proposed a Deep Network (DN) architecture called Learned Invariant Feature Transform (LIFT) which implements a detection system, an orientation estimation and a feature description. Fig. 12 shows the LIFT system overview diagram.

The Descriptor for LIFT can be defined as:

$$d = h_\rho(P_\theta), \tag{13}$$

where h, ρ and P_θ are the Descriptor CNN, its parameters, and the rotated patch from the Orientation Estimator, respectively. This method is a fast and real time deep learning system which can be used in SLAM techniques.

Alahi, Ortiz, and Vandergheynst (2012) proposed a novel method based on the human visual system and retina called Fast Retina Keypoint (FREAK). This algorithm is a binary feature method. By changing the size of the Gaussian kernels, FREAK builds a system which is robust to orientation, noise, and scale with low calculation complexity. The FREAK descriptor can be defined as a sequence of one-bit Difference of Gaussians (DoG):

$$F = \sum_{0 \le \alpha \le N} 2^\alpha T(P_\alpha) \quad where \quad T(P_\alpha) = \begin{cases} 1 I\left(P_\alpha^{\gamma_1}\right) > I\left(P_\alpha^{\gamma_2}\right) \\ 0 Otherwise \end{cases}, \tag{14}$$

where N, P_α and $I(P_\alpha^{\gamma 1})$ are the size of the descriptor, a pair of receptive fields and the smoothed intensity of the first receptive field of P_α. Fig. 13 shows a FREAK image matching result for a sample image.

Leutenegger, Chli, and Siegwart (2011) proposed a technique for key-point feature detection, description and image matching called Binary Robust Invariant Scalable Key-points (BRISK). The BRISK algorithm is a combination of the BRIEF descriptor and the SIFT scale key-point detector. To build a scale and rotation invariant normalized descriptor, the sampling pattern rotates by atan2(g_y, g_x) around the key-point k (g_x and g_y are gradients sum of the long-distance point 2D pair). This method is a 512-bit binary descriptor and like other binary descriptors works with hamming distance instead of Euclidean distance. Fig. 14 shows results of the use of the BRISK algorithm for feature extraction and image matching.

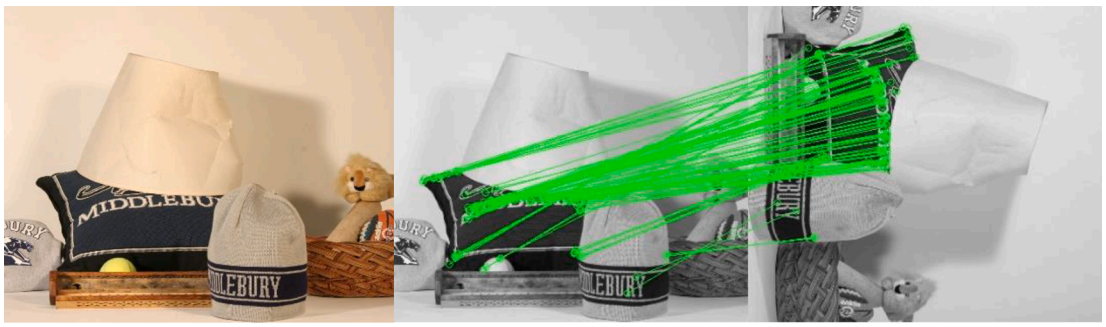
Mikolajczyk and Schmid (2005) presented a robust descriptor called Gradient Location and Orientation Histogram (GLOH). GLOH is an extension of the SIFT algorithm which integrates position information and local appearance by considering more spatial regions for the histograms. This technique computes the SIFT algorithm for a log-polar location grid and generates a 272-bin histogram and then using a PCA technique, the size of the results is reduced.

Tola, Lepetit, and Fua (2009) presented a local image descriptor based on the GLOH and SIFT algorithms for dense and large images called A Fast-Local Descriptor for Dense Matching (DAISY). This technique is robust to contrast, scale, rotation, illumination, and perspective changes. This algorithm utilizes the convolutions of the original image with several oriented derivatives of Gaussian filters with large standard deviations. The DAISY descriptor can be defined as:

$$\begin{aligned} D(u_0, v_0) = \Big[& \tilde{h}^T_{\sum_1}(u_0, v_0), \\ & \tilde{h}^T_{\sum_1}(I_1(u_0, v_0, R_1)), \cdots, \tilde{h}^T_{\sum_1}(I_N(u_0, v_0, R_1)), \\ & \tilde{h}^T_{\sum_2}(I_1(u_0, v_0, R_2)), \cdots, \tilde{h}^T_{\sum_2}(I_N(u_0, v_0, R_2)), \\ & \tilde{h}^T_{\sum_3}(I_1(u_0, v_0, R_3)), \cdots, \tilde{h}^T_{\sum_3}(I_N(u_0, v_0, R_3)) \Big]^T, \end{aligned} \tag{15}$$

where h is the vector made of the values at (u, v) in the orientation maps after convolution by a Gaussian kernel of standard deviation Σ and I_j is the location with distance R from pixel (u, v) in the direction given by j when the directions are quantized in N values. This algorithm is good for noisy environments with changing light. Fig. 15 shows results using the DAISY algorithm for feature extraction and image matching.

Wang, Fan, and Wu (2011) presented a method for feature description based on intensity order called Local Intensity Order Pattern (LIOP).



(a) (b)

Fig. 11. ORB algorithm (a) Original image and (b) Image matching for original image and 90 degrees rotated original image.

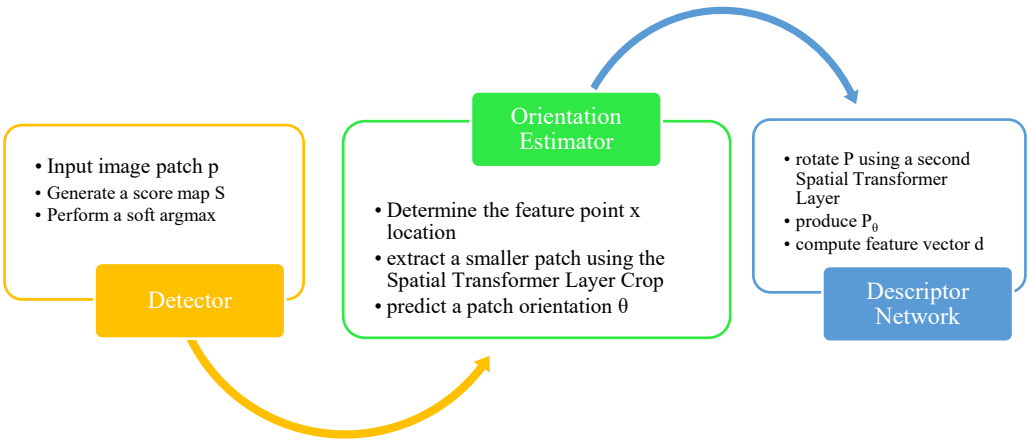
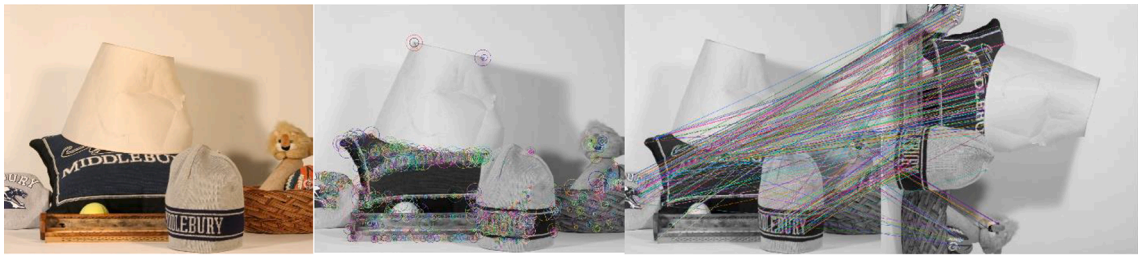


Fig. 12. LIFT overview, showing all the steps including detector, orientation estimator and descriptor.

(a) (b)

Fig. 13. FREAK algorithm (a) Original image and (b) Image matching for original image and 90 degrees rotated original image.



(a) (b) (c)

Fig. 14. Brisk algorithm (a) Original image, (b) key-point detection (gray-scale level) and (c) Image matching for original image and 90 degrees rotated original image.

(a) (b) (c)

Fig. 15. DAISY algorithm (a) Original image, (b) key-point detection (gray-scale level) and (c) Image matching for original image and 90 degrees rotated original image.

In this algorithm local and overall intensity ordinal information of the local patch are considered to make a discriminative descriptor. LIOP is invariant to image blur, image rotation and monotonic intensity changes and can be briefly defined as:

$$LIOP = \sum_{x \in bin_i} V_{N!}^{Ind(\gamma(P(x)))}, \tag{16}$$

where P(x) is the intensity of the i-th of N neighboring sample point x_i and Ind is the index of mapping γ (π or 0) obtained based on the proposed LIOP index table. Large computation cost makes this method complex for real time applications.

Fan, Wu, and Hu (2011) present a local image descriptor that combines gradient distributions and intensity orders in multiple support regions Called MROGH. In the MROGH algorithm, the gradient is obtained in each support region and then the rotation invariant gradients are pooled spatially based on intensity orders and finally multiple support regions are used to create an MROGH descriptor. A rotation invariant gradient which is the fundamental component of this technique can be defined as:

$$\begin{cases} D_x(P_i) = I(P_i^1) - I(P_i^3) \\ D_y(P_i) = I(P_i^2) - I(P_i^4) \end{cases}, \tag{17}$$

where p_i^n is n-th neighboring points of candidate pixel P_i for (x, y) coordinate and I is the intensity of that point. In this algorithm, intensity orders of candidate points are considered instead of geometric locations.

Xu, Tian, Feng, and Zhou (2014) proposed a binary descriptor based on the ordinal and spatial information of regional invariants called OSRI. To create a fast and low-cost computation method, the OSRI builds a subregion set from a region by designing rotation invariant sampling and according to high level invariants defines a descriptor. The spatial order invariants of geometric centroids of subregions are defined based on the ORB method. To obtain the spatial distribution information of points in subregions, each bin of the histogram can be defined as:

$$\Phi_{ij} = \frac{1}{2j-1} \left| \{ (x,y) | (x,y) \in Sub\, R_i \,\&\, (x,y) \in \mathbb{O}_j \} \right|, \tag{18}$$

where Sub R_i is the i-th subregion, (x,y) is a point, $|\cdot|$ is the cardinality of a set, $\mathbb{O}_j$ is the j-th concentric ring in a support region. By comparing a small portion of the descriptor, this algorithm rejects non-matching descriptors at early stages.

Han, Leung, Jia, Sukthankar, and Berg (2015) proposed Match-Net: a unified approach to combine learning feature representations and comparison functions for training a patch matching system. This system is a deep convolutional network for patch feature extraction and a network of three fully connected layers to compute similarity of these extracted features. Match-Net is useful for wide-baseline viewpoint invariant matching. This technique utilizes Rectified Linear Units (ReLU) for non-linearity for the convolution layers and minimizes the cross-entropy error over a training set of n patch pairs using Stochastic Gradient Descent (SGD) with a batch size of 32 which can be defined as:

$$E = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)], \tag{19}$$

Here y_i is the label of input pair x_i and $\hat{y}_i$ is the SoftMax activation and is used as the probability estimate for label 1. This technique is a proper deep learning approach for image matching and object detection.

Alcantarilla, Bartoli, and Davison (2012) presented a multi-scale 2D algorithm in nonlinear scale spaces called KAZE. This algorithm is based on a scale normalized determinant of the Hessian Matrix. The KAZE features are invariant to scale and rotation. This method calculates the main orientation of the key-point and computes a rotation and scale invariant descriptor using first order image derivatives. To achieve a set of evolution times (t_i) and transform the scale space to time units, the nonlinear scale space can be defined as:

$$L^{i+1} = \left(I - \tau \sum_{l=1}^{m} A_l(L^i) \right)^{-1} L^i \quad where \quad \tau = (t_{i+1} - t_i), \tag{20}$$

where L is image luminance and A_l is a matrix which encodes the image conductivities for each dimension. The nonlinear scale space in this algorithm has been built using variable conductance diffusion and efficient Additive Operator Splitting (AOS) techniques.

Alcantarilla and Solutions (2011) also, proposed a fast multi-scale feature detection approach that exploits the aids of nonlinear scale spaces called Accelerated-KAZE (A-KAZE). Using preprocessing techniques before the main algorithms is often useful for more accurate results. In SLAM, image preprocessing techniques such as noise removal and enhancement (especially in underwater SLAM applications) can reduce navigation and path finding errors. It is to be noted that using these techniques can affect processing times, but there is a trade-off between processing times and accuracy. It is also important to be aware that noise reduction techniques can potentially damage the main or important pixels in the image. This method utilizes the non-linear scale space that blurs the image pixels, resulting in a noise removal process without damaging the main image pixels. Using the fast-explicit diffusion (FED) algorithm and the principle of a nonlinear diffusion filter, a non-linear scale space is built. The image luminance is diffused by the non-linear nature of the partial differential equation of the nonlinear scale space. In this algorithm, after feature extraction, the element of the Hessian for each of the filtered images Lum^i in the nonlinear scale space will be computed. The calculation of the Hessian matrix can be defined by:

$$H(Lum^i) = \sigma_{i,norm}^2 \left(Lum_{xx}^i Lum_{yy}^i - Lum_{xy}^i Lum_{xy}^i \right), \tag{21}$$

where $\sigma_{i,norm}^2$ is the normalized scale factor of the octave of each image in the nonlinear scale (i.e. $\sigma_{i,norm} = \sigma_i / 2^{\sigma_i}$). Lum_{xx}^i and Lum_{yy}^i are the horizontal and vertical image of the second-order derivative, respectively and Lum_{xy}^i is the cross-partial derivative. The eigenvectors with scale and rotation invariance are extracted based on the first-order differential images. The A-KAZE algorithm uses a Modified-Local Difference Binary (M−LDB) to describe the feature points and exploit gradient and intensity information from the nonlinear scale space. Fig. 16 shows the A-KAZE algorithm result for an image matching application.

Yang and Cheng (2012) introduced a binary descriptor called Local Difference Binary (LDB) and developed the same principle as BRIEF (Calonder et al., 2011). This algorithm using gradient and intensity difference among pairwise grid cells, calculates a binary string for an image patch. LDB divides each image patch P into n*n grids and computes a binary test as:

$$\tau(F(i), F(j)) = \begin{cases} 1 F(i) > F(j) \quad and \quad i \neq j \\ 0 Otherwise \end{cases}, \tag{22}$$

where F is the function for extracting information from a grid cell. LDB computes the intensity summation and gradients of any grid cell based on the rotated integral image of the patch.

5. Deep learning

Deep Learning (DL) is a training Artificial Neural Network (ANN) technique with a large number of hidden layers between the input and output layers. Deep Neural Network (DNN) is a basic supervised DL model organized in a layer-wise structure. Convolutional Neural Networks (CNNs) or ConvNet is a deep neural network that uses convolutional and pooling layers and is currently used widely in computer vision, image classification and robotic systems. This popular technique uses convolution filter structure for feature extraction in 1D, 2D and 3D matrix data and images as input. CNN is the main core of most neural

(a) (b)

Fig. 16. A_KAZE algorithm (a) Original image and (b) Image matching for original image and 90 degrees rotated original image.

network algorithms and can be trained in a supervised or unsupervised manner. For SLAM approaches, there are several DL techniques used for system learning such as Deep Recurrent Convolutional Neural Network (RCNN), deep Recurrent Neural Networks (RNN), Generative Adversarial Network (GAN), etc. and currently, several deep learning systems are proposed based on CNN algorithms. Fig. 17 shows a symbolic CNN based network for 3D imaging. Input, output and multiple convolutional layers (pooling, fully connected (FC) and normalization layers) are the main parts of a CNN system.

DL can be used in the field of V-SLAM techniques and can optimize Visual Odometry and Loop Closure sections. Recently, deep learning camera motion and pose estimation techniques, are widely used in mobile robot systems. Such systems can replace the full traditional V-SLAM pipeline that need substantial engineering effort for developing. Feature extraction and matching are the core of feature-based VO algorithms and DL techniques use these basic algorithms to build neural network systems. LIBVISO (Kitt, Geiger, & Lategahn, 2010) is a library for stereo and monocular images for VO application and 6-DoF pose estimation.

The Bag-of-Words (BoW) method is one of the popular algorithms for loop detection in mobile robot SLAM. BoW generates a visual dictionary based on objects and scenes visual data called words and uses an inverted index to retrieve previously seen scenes and objects (Garcia-Fidalgo & Ortiz, 2018). Feature extraction and matching discussed in the previous section of this paper are based on this method. This technique needs a large amount of memory to process the data, and this is an important limitation for robotic systems. The FAB-MAP (Cummins & Newman, 2011) system is built based on this technique to check the new image features and compare them with the prepared dictionary.

DL can improve this method by adding a training data phase to the system before loop closure (Memon, Wang, & Hussain, 2020). Thus, the mobile robot knows and understands the environment objects and scenes during loop closure detection. Table 3 shows the DL methods used in the state-of-the-art literature and reports for visual odometry and loop closure (VO and LC) in the SLAM techniques.

6. Datasets

Unlike traditional machine vision methods, neural networks require data with ground truth for optimizing parameters and to learn to perform the task. Thus, it is necessary to train and test the DL method on datasets which provide images and ground truth as network inputs. There are variety valid datasets for SLAM approaches for training and testing.

The KITTI (Geiger, Lenz, Stiller, & Urtasun, 2013) dataset contains image pairs captured from a mobile vehicle with sequences of outdoor real scenes and ground truth obtained by GPS and a Velodyne laser scanner. This dataset and its update with more moving objects (Menze &

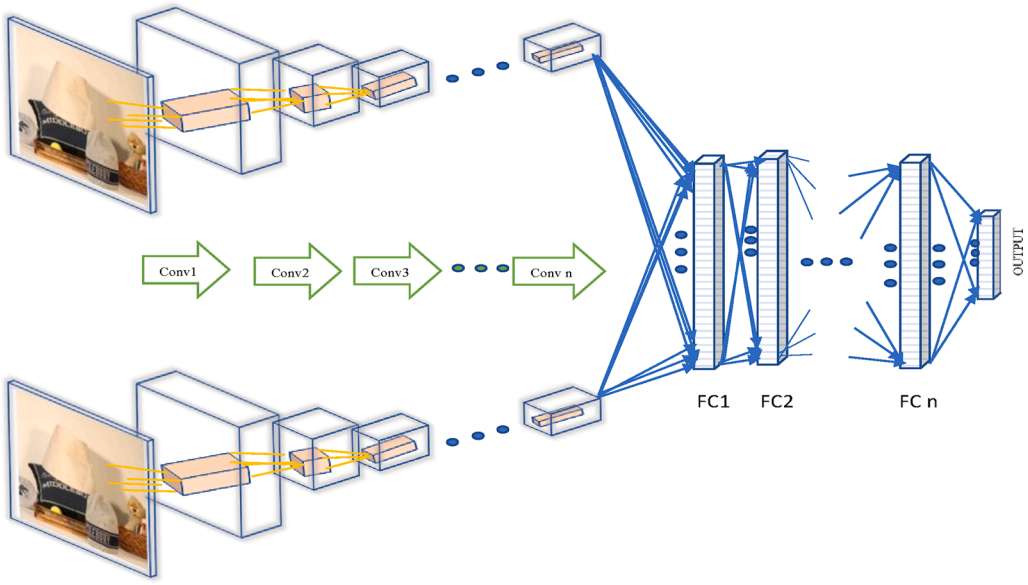


Fig. 17. Structure of a typical 3D Convolutional Neural Network (CNN).

Table 3
The DL methods used in the state-of-the-art literature.

Ref.	VO/ LC	Manner	Method	Dataset	Framework	System
(Mohanty, Agrawal, Datta, Ghosh, Sharma, & Chakravarty, 2016)	VO	Supervised	CNN based Deep VO	KITTI	Caffe	Intel Xeon @4 × 3.3 GHZ with 32 GB DDR3 RAM and NVIDIA GTX 970
(Costante, Mancini, Valigi, & Ciarfuglia, 2015)	VO	Supervised	CNN-1b VO, CNN-4b VO and P-CNN VO	KITTI	NA	NVIDIA Tesla K40
(Wang, Clark, Wen, & Trigoni, 2017)	VO	Supervised	Deep RCNNs-DeepVO	KITTI	NA	NVIDIA TeslaK40 GPU
(Melekhov, Ylioinas, Kannala, & Rahtu, 2017)	VO	Supervised	CNN-SPP (spatial pyramid pooling (SPP))	DTU	Torch	NVIDIA Titan X
(Turan, Almalioglu, Araujo, Konukoglu, & Sitti, 2018)	VO	Supervised	RCNN- Deep EndoVO	Five different real pig stomachs	Caffe	NVIDIA Tesla K40 GPU
(Li, Wang, Long, & Gu, 2018)	VO	Supervised	UnDeepVO	KITTI	TensorFlow	NVIDIA Tesla P100 GPU and NVIDIA GeForce GTX 980 M
(Clark, Wang, Wen, Markham, & Trigoni, 2017)	VO	Supervised	CNN-RNN based VINet	EuRoC Micro Aerial Vehicle (MAV) (Burri et al., 2016) and KITTI	Theano	Tesla k80
(Kang, Shi, Li, Liu, & Liu, 2019)	VO	Supervised	CNN-Based DF-SLAM	TUM, EuRoC	Pytorch	GeForce GTX TITAN X/PCIe/ SSE2
(Peretroukhin, Wagstaff, Giamou, & Kelly, 2019)	VO	Supervised	Hydra Net	KITTI	Pytorch	NA
(Guclu, Caglayan, & Burak Can, 2019)	VO	Supervised	CNN-RNN	TUM	NA	Intel Core i7-2600 CPU at 3.40 GHz and 8 GB RAM
(Teixeira, Silva, Matos, & Silva, 2020)	VO	Supervised	RCNN	Data acquired with the UX-1 robot	TensorFlow on CUDA	Nvidia GTX 1080
(Zhou, Brown, Snavely, & Lowe, 2017)	VO	Unsupervised	SfM Learner (Single-view depth andMulti-view pose networks)	Cityscapes + KITTI to the Make3D	TensorFlow	NA
(Vijayanarasimhan, Ricco, Schmid, Sukthankar, & Fragkiadaki, 2017)	VO	Unsupervised	SfM-Net	KITTI, MoSeg and TUM	Python	NA
(Mahjourian, Wicke, & Angelova, 2018)	VO	Unsupervised	SfM Learner architecture and Disp Net	KITTI and Uncalibrated Bike Video Dataset	TensorFlow	NA
(Zhan et al., 2018)	VO	Unsupervised	Depth Conv Net(CNND) and VO Conv Net (CNNVO)	KITTI	Caffe	NA
(Yin & Shi, 2018)	VO	Unsupervised	Geo Net	KITTI	TensorFlow	Titan XP GPU
(Prasad & Bhowmick, 2019)	VO	Unsupervised	SfM Learner++	KITTI, Cityscapes and Make3D	TensorFlow	NA
(Luo, Yang, Wang, Wang, Xu, Nevatia, & Yuille, 2018)	VO	Unsupervised	Every Pixel Counts++ (EPC++)	KITTI, Make3D and MPI-Sintel (Butler et al., 2012)	TensorFlow	NA
(Zhu, Liu, Wang, Kumar, & Daniilidis, 2018)	VO	Unsupervised	Flownet-S (Dosovitskiy et al., 2015) based architecture	KITTI	NA	NA
(Valada, Radwan, & Burgard, 2018)	VO	Unsupervised	DCNN based VLocNet	Microsoft 7-Scenes(Shotton et al., 2013) and Cambridge Landmarks (Kendall, Grimes, & Cipolla, 2015)	TensorFlow	NVIDIA Titan X GPU
(Radwan et al., 2018)	VO	Unsupervised	VLocNet++	Microsoft 7-Scenes and DeepLoc	TensorFlow	NVIDIA Titan X GPU
(Han, Lin, Du, & Lian, 2019)	VO	Unsupervised	CNN-Flow based DeepVIO	KITTI and EuRoC MAV	NA	Nvidia GeForce GTX1080 Ti with 12G memory
(Almalioglu, Saputra, de Gusmao, Markham, & Trigoni, 2019)	VO	Unsupervised	Deep Convolutional Generative Adversarial Networks (GANs)(GANVO)	Cityscapes (Cordts et al., 2016) and KITTI	TensorFlow	NVIDIA TITAN V model GPU
(Feng & Gu, 2019)	VO	Unsupervised	Stacked Generative Adversarial Networks (SGANVO)	KITTI	TensorFlow	NVIDIA GTX 1080TI GPU
(Xiao, Wang, Qiu, Rong, & Zou, 2019)	VO	Unsupervised	CNN based Dynamic-SLAM	TUM and KITTI and real time data	C++	Intel Core i5-7300HQ CPU,8GB RAM and NVIDIA GTX1050Ti and VIDIA Jetson TX2
(Naseer, Ruhnke, Stachniss, Spinello, & Burgard, 2015)	LC	Supervised	Deep Convolutional NeuralNetworks (DCNN)	Two authors-built dataset and City Centre and New College	NA	NA
(Bai, Wang, Zhangt, Yi, & Tang, 2016)	LC	Supervised	CNN	City Centre and New College	Caffe- Python	3.60 GHz CPU with four cores and 8 GB memory
(Zhang, Su, & Zhu, 2017)	LC	Supervised	CNN and CNN-Aug	City Centre and New College	NA	NA
(Gao & Zhang, 2015)	LC	Unsupervised	Stacked auto-encoder	Freiburg 2(Sturm et al., 2012)	Theano	NA
(Gao & Zhang, 2017)	LC	Unsupervised	Stacked Denoising Auto-encoder (SDA)	Fr_Office, City Centre and New College	Theano	NA
(Wang, Peng, Guan, & Wu, 2019)	LC	Unsupervised	Graph-Regularization Stacked Denoising Auto-encoder (G-SDAE)	City Centre and New College, Fr3_Office and Fr3_Texture (Smith, Baldwin, Churchill, Paul,	Theano	DELL OPTIPLEX 9020

(continued on next page)

Table 3 (*continued*)

Ref.	VO/ LC	Manner	Method	Dataset	Framework	System
				& Newman, 2009; Sturm et al., 2012)		
(Memon et al., 2020)	LC	Supervised and Unsupervised	CNN and Autoencoder	City Centre, KITTI and Gardens Point Walk (Merrill & Huang, 2018)	NA	Intel Core i5-3470CPU with 8 GB RAM and NVIDIA Geforce GTX 770
(Ramezani, Tinchev, Iuganov, & Fallon, 2020)	LC	Unsupervised	Point CNN-Efficient Segment Matching (ESM)(Tinchev, Penate-Sanchez, & Fallon, 2019)	Collected by ANYmal quadruped robot	TensorFlow	NA
(Li, Wang, & Gu, 2020)	LC	Unsupervised	RCNN based Deep SLAM	KITTI	TensorFlow	NVIDIA DGX-1 with TeslaP100 (Train)- NVIDIA GeForce GTX 980 M GPU and Intel Core i7-6820HK 2.7 GHz CPU(Test)
(Mukherjee, Chakraborty, & Saha, 2019)	LC	Unsupervised	DeConvNet	KITTI	Theano	NVidia Quadro M5000 GPU with 8 GB GPU memory
(Liu, Suo, Zhou, Wei, Liu, Wang, & Liu, 2019)	LC	Unsupervised	SeqLPD (large-scale place description network (LPD-Net))	Oxford RobotCar (Maddern, Pascoe, Linegar, & Newman, 2017)	TensorFlow	NVIDIA 1080Ti GPU, i7-6700 processor, and 16G RAM

Geiger, 2015) are large computer vision datasets for use with mobile robots' algorithms and contain 200 stereo pairs and frame sequences (approximately 42,000 frames). The dataset is a big data source for stereo video, 3D point clouds from LIDAR and vehicle trajectory.

MoSeg dataset (Brox & Malik, 2010) contains sequences with challenging object motion, including articulated motions from moving animals and people. The MPI Sintel (Butler, Wulff, Stanley, & Black, 2012) dataset is another dataset with 1041 training image pairs.

ImageNet (Deng et al., 2009) dataset is an image dataset for training network systems which contains 1.2 million images of 1000 categories. Usually, networks train on this database for feature extraction and object detection in many applications.

DTU Robot Image Dataset (Jensen, Dahl, Vogiatzis, Tola, & Aanæs, 2014) covers 124 scenes. It includes 77 scenes (type-I) with 49 camera positions and 47 scenes (type-II) with 64 camera positions.

The TUM RGB-D SLAM benchmark (Sturm et al., 2012) datasets are collected by the TUM computer vision group and is a good dataset for dynamic SLAM methods.

The City Centre and New College datasets (Cummins & Newman, 2008) contain 2474 and 2146 images respectively with 640 × 480 pixels. Images are outdoor urban environments with stable lighting conditions, and they are numbered in the order of collection. The images were captured using two cameras mounted on a pan-tilt to collect images from the left and right of the robot. The ground-truth loop closures for the City Centre and New College datasets are 26,976 and 14832, respectively.

7. Conclusion

The knowledge of Visual SLAM is valuable for smart robotic applications. This article has presented a review of the state-of-the-art V-SLAM methods, as well as reviewing comparisons and applicability of methods based on their intended environment. There currently is no one size fits all approach that can be applied to any use case. Some methods may require greater computational resources while offering a more detailed and denser map of large environments. Others may depend on sparse features, offering quick localization and mapping due to the speed and robustness of modern feature detectors and descriptors, allowing for good operation on mobile devices with limited resources.

Also, traditional and modern SLAM models, sensors, feature-based algorithms and new methods have been presented and some of these algorithms have been simulated experimentally to show the results of each method. Currently, deep learning visual SLAM techniques are used to reduce computation time and increase accuracy. We have summarized a range of DL techniques, frameworks, processors, and datasets used in the mobile robot vision research area. In this study, feature extraction and matching algorithms in V-SLAM have been discussed and for future works, the object detection and moving objects removal techniques in V-SLAM can be studied.

Visual SLAM technology needs to overcome the problems of large computation for large scale environments, motion distortion and balancing the precision and real-time process relationship. This was identified as an early problem in SLAM systems, notably being seen with Mono-SLAM, and has been worked on subsequently but is still not completely solved. Despite fascinating progress over the past years, existing SLAM systems are far from building actionable and complete maps and models of the unknown environment, comparable to human built maps. Another problem with area for improvement is the semantic based approach. Autonomous robots could function more intelligently if they had a better meaningful understanding of their surroundings, especially in unstructured outdoor environments.

Deep Learning has provided breakthroughs in many academic and industrial sectors over recent years, particularly for objectives such as decision making and image analysis and manipulation, offering a great presentation of autonomy and speed. This is seen as a way forward for V-SLAM systems and SLAM in general. State-of-the-art deep learning techniques can be used to improve SLAM processes including loop closure, data processing, pose estimation, trajectory, and mapping to make a high accuracy and fast SLAM in the future. Large-scale datasets and their availability are important for deep learning applications but can also be seen as a limiting factor when the data is scarce or unlabeled. The attempt to build well-functioning unsupervised learning models is promising.

Funding

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement, SMART 4.0, No. 847577; and a research grant from Science Foundation Ireland (SFI) under Grant Number 16/RC/3918 (Ireland's European Structural and Investment Funds Programmes and the European Regional Development Fund 2014-2020).

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abaspur Kazerouni, I., Dooly, G., & Toal, D. (2020). Underwater Image Enhancement and Mosaicking System Based on A-KAZE Feature Matching. *Journal of Marine Science and Engineering, 8*(6), 449.

Ajay, A., & Venkataraman, D. (2013). A survey on sensing methods and feature extraction algorithms for SLAM problem. *arXiv preprint arXiv:1303.3605*.

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). Freak: Fast retina keypoint. *Paper presented at the 2012 IEEE Conference on Computer Vision and Pattern Recognition*.

Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012). KAZE features. *Paper presented at the European Conference on Computer Vision*.

Alcantarilla, P. F., & Solutions, T. (2011). Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(7), 1281–1298.

Almalioglu, Y., Saputra, M. R. U., de Gusmao, P. P., Markham, A., & Trigoni, N. (2019). Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. *Paper presented at the 2019 International Conference on Robotics and Automation (ICRA)*.

Bai, D., Wang, C., Zhangt, B., Yi, X., & Tang, Y. (2016). Matching-range-constrained real-time loop closure detection with CNNs features. *Paper presented at the 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*.

Balakrishnama, S., & Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. *Paper presented at the Institute for Signal and information Processing*.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding, 110*(3), 346–359.

Brox, T., & Malik, J. (2010). Object segmentation by long term analysis of point trajectories. *Paper presented at the European conference on computer vision*.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., … Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research, 35*(10), 1157–1163.

Butler, D. J., Wulff, J., Stanley, G. B., & Black, M. J. (2012). A naturalistic open source movie for optical flow evaluation. *Paper presented at the European conference on computer vision*.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., … Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics, 32*(6), 1309–1332.

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., & Fua, P. (2011). BRIEF: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(7), 1281–1298.

Chatila, R., & Laumond, J.-P. (1985). Position referencing and consistent world modeling for mobile robots. *Paper presented at the Proceedings. 1985 IEEE International Conference on Robotics and Automation*.

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing, 10*(3), 145–155.

Chen, Y., Zhou, Y., Lv, Q., & Deveerasetty, K. K. (2018). A Review of V-SLAM. *Paper presented at the 2018 IEEE International Conference on Information and Automation (ICIA)*.

Chong, T., Tang, X., Leng, C., Yogeswaran, M., Ng, O., & Chong, Y. (2015). Sensor technologies and simultaneous localization and mapping (SLAM). *Procedia Computer Science, 76*, 174–179.

Civera, J., Davison, A. J., & Montiel, J. M. (2008). Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics, 24*(5), 932–945.

Clark, R., Wang, S., Wen, H., Markham, A., & Trigoni, N. (2017). Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. *Paper presented at the Thirty-First AAAI Conference on Artificial Intelligence*.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., … Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. *Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition*.

Costante, G., Mancini, M., Valigi, P., & Ciarfuglia, T. A. (2015). Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE Robotics and Automation Letters, 1*(1), 18–25.

Cummins, M., & Newman, P. (2008). FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research, 27*(6), 647–665.

Cummins, M., & Newman, P. (2011). Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research, 30*(9), 1100–1123.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. *Paper presented at the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*.

Davison, A. J. (2003). Real-time Simultaneous localization and mapping with a single camera. Retrieved from *IEEE International Conference on Computer Vision*, 1403–1410 https://ci.nii.ac.jp/naid/10025007139/en/.

Debeunne, C., & Vivet, D. (2020). A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping. *Sensors, 20*(7), 2068.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *Paper presented at the 2009 IEEE conference on computer vision and pattern recognition*.

Dewi, D. A., Sundararajan, E., Prabuwono, A. S., & Cheng, L. M. (2019). Object detection without color feature: Case study autonomous robot. *International Journal of Mechanical Engineering and Robotics Research, 8*(4), 646–650.

Donoso, F., Austin, K. J., & McAree, P. R. (2017). Three new Iterative Closest Point variant-methods that improve scan matching for surface mining terrain. *Robotics and Autonomous Systems, 95*, 117–128.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., … Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. *Paper presented at the Proceedings of the IEEE international conference on computer vision*.

Dryanovski, I., Valenti, R. G., & Xiao, J. (2013). Fast visual odometry and mapping from RGB-D data. *Paper presented at the 2013 IEEE international conference on robotics and automation*.

Duan, C., Junginger, S., Huang, J., Jin, K., & Thurow, K. (2019). Deep Learning for Visual SLAM in Transportation Robotics: A review. *Transportation Safety and Environment, 1*(3), 177–184.

Dubé, R., Sommer, H., Gawel, A., Bosse, M., & Siegwart, R. (2016). Non-uniform sampling strategies for continuous correction based trajectory estimation. *Paper presented at the 2016 IEEE International Conference on Robotics and Automation (ICRA)*.

Engel, J., Koltun, V., & Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(3), 611–625.

Engel, J., Schöps, T., & Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. *Paper presented at the European conference on computer vision*.

Fan, B., Wu, F., & Hu, Z. (2011). *Aggregating gradient distributions into intensity orders: A novel local image descriptor*. Paper presented at the CVPR 2011.

Feng, T., & Gu, D. (2019). Sganvo: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks. *IEEE Robotics and Automation Letters, 4*(4), 4431–4437.

Gao, B., Lang, H., & Ren, J. (2020). Stereo Visual SLAM for Autonomous Vehicles: A Review. *Paper presented at the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*.

Gao, X., & Zhang, T. (2015). Loop closure detection for visual slam systems using deep neural networks. *Paper presented at the 2015 34th Chinese Control Conference (CCC)*.

Gao, X., & Zhang, T. (2017). Unsupervised learning to detect loops using deep neural networks for visual SLAM system. *Autonomous Robots, 41*(1), 1–18.

Garcia-Fidalgo, E., & Ortiz, A. (2018). ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words. *IEEE Robotics and Automation Letters, 3*(4), 3051–3057.

Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research, 32*(11), 1231–1237.

Gokulalakshmi, A., Karthik, S., Karthikeyan, N., & Kavitha, M. (2020). ICM-BTD: Improved classification model for brain tumor diagnosis using discrete wavelet transform-based feature extraction and SVM classifier. *Soft Computing*, 1–11.

Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics, 23*(1), 34–46.

Guclu, O., Caglayan, A., & Burak Can, A. (2019). RGB-D Indoor Mapping Using Deep Features. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Han, L., Lin, Y., Du, G., & Lian, S. (2019). DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3D geometric constraints. *arXiv preprint arXiv:1906.11435*.

Han, X., Leung, T., Jia, Y., Sukthankar, R., & Berg, A. C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hess, W., Kohler, D., Rapp, H., & Andor, D. (2016). Real-time loop closure in 2D LIDAR SLAM. *Paper presented at the 2016 IEEE International Conference on Robotics and Automation (ICRA)*.

Jamiruddin, R., Sari, A. O., Shabbir, J., & Anwer, T. (2018). RGB-depth SLAM review. *arXiv preprint arXiv:1805.07696*.

Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., & Aanæs, H. (2014). Large scale multi-view stereopsis evaluation. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kaess, M., Ranganathan, A., & Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics, 24*(6), 1365–1378.

Kang, R., Shi, J., Li, X., Liu, Y., & Liu, X. (2019). DF-SLAM: A deep-learning enhanced visual SLAM system based on deep local features. *arXiv preprint arXiv:1901.07223*.

Kazerouni, I., & Haddadnia, J. (2014). A mass classification and image retrieval model for mammograms. *The Imaging Science Journal, 62*(7), 353–357.

Keller, M., Lefloch, D., Lambers, M., Izadi, S., Weyrich, T., & Kolb, A. (2013). Real-time 3d reconstruction in dynamic scenes using point-based fusion. *Paper presented at the 2013 International Conference on 3D Vision-3DV 2013*.

Kendall, A., Grimes, M., & Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. *Paper presented at the Proceedings of the IEEE international conference on computer vision*.

Kitt, B., Geiger, A., & Lategahn, H. (2010). Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. *Paper presented at the 2010 ieee intelligent vehicles symposium*.

Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. *Paper presented at the 2007 6th IEEE and ACM international symposium on mixed and augmented reality*.

Kohlbrecher, S., Von Stryk, O., Meyer, J., & Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. *Paper presented at the 2011 IEEE international symposium on safety, security, and rescue robotics*.

Kolhatkar, C., & Wagle, K. Review of SLAM Algorithms for Indoor Mobile Robot with LIDAR and RGB-D Camera Technology. In *Innovations in Electrical and Electronic Engineering* (pp. 397-409): Springer.

Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., & Vincent, R. (2010). Efficient sparse pose adjustment for 2D mapping. *Paper presented at the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Lee, T. S. (1996). Image representation using 2D Gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(10), 959–971.

Leonard, J. J., & Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. *Paper presented at the IROS*.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. *Paper presented at the 2011 International conference on computer vision*.

Li, R., Wang, S., & Gu, D. (2020). DeepSLAM: A Robust Monocular SLAM System with Unsupervised Deep Learning. *IEEE Transactions on Industrial Electronics*.

Li, R., Wang, S., Long, Z., & Gu, D. (2018). Undeepvo: Monocular visual odometry through unsupervised deep learning. *Paper presented at the 2018 IEEE international conference on robotics and automation (ICRA)*.

Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A., Daniilidis, K., … Asfour, T. (2020). *TLIO: Tight Learned Inertial Odometry*. IEEE Robotics and Automation Letters.

Liu, Z., Suo, C., Zhou, S., Wei, H., Liu, Y., Wang, H., & Liu, Y.-H. (2019). SeqLPD: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles. *arXiv preprint arXiv:1904.13030*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110.

Lowry, S., Sünderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., & Milford, M. J. (2015). Visual place recognition: A survey. *IEEE Transactions on Robotics, 32*(1), 1–19.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots, 4*(4), 333–349.

Luo, C., Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R., & Yuille, A. (2018). Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*.

Maddern, W., Pascoe, G., Linegar, C., & Newman, P. (2017). 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research, 36*(1), 3–15.

Mahjourian, R., Wicke, M., & Angelova, A. (2018). Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

McCormac, B. J. (2018). *SLAM and deep learning for 3D indoor scene understanding*. Imperial College London.

Melekhov, I., Ylioinas, J., Kannala, J., & Rahtu, E. (2017). Relative camera pose estimation using convolutional neural networks. *Paper presented at the International Conference on Advanced Concepts for Intelligent Vision Systems*.

Memon, A. R., Wang, H., & Hussain, A. (2020). Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems. *Robotics and Autonomous Systems, 126*, Article 103470.

Menze, M., & Geiger, A. (2015). Object scene flow for autonomous vehicles. *Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition*.

Merrill, N., & Huang, G. (2018). Lightweight unsupervised deep loop closure. *arXiv preprint arXiv:1805.07703*.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(10), 1615–1630.

Milford, M. J., Wyeth, G. F., & Prasser, D. (2004). *RatSLAM: a hippocampal model for simultaneous localization and mapping*. In Paper presented at the IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004.

Milford, M. J., & Wyeth, G. F. (2012). SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. *Paper presented at the 2012 IEEE International Conference on Robotics and Automation*.

Mohanty, V., Agrawal, S., Datta, S., Ghosh, A., Sharma, V. D., & Chakravarty, D. (2016). Deepvo: A deep learning approach for monocular visual odometry. *arXiv preprint arXiv:1611.06069*.

Montemerlo, M., Thrun, S., Koller, D., & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai, 593598*.

Mukherjee, A., Chakraborty, S., & Saha, S. K. (2019). Detection of loop closure in SLAM: A DeconvNet based approach. *Applied Soft Computing, 80*, 650–656.

Murangira, A., Musso, C., & Dahia, K. (2016). A mixture regularized rao-blackwellized particle filter for terrain positioning. *IEEE Transactions on Aerospace and Electronic Systems, 52*(4), 1967–1985.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics, 31*(5), 1147–1163.

Mur-Artal, R., & Tardós, 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.

Naseer, T., Ruhnke, M., Stachniss, C., Spinello, L., & Burgard, W. (2015). Robust visual SLAM across seasons. *Paper presented at the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., … Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. *Paper presented at the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*.

Newcombe, R. A., Lovegrove, S. J., & Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. *Paper presented at the 2011 international conference on computer vision*.

Ondrúška, P., Kohli, P., & Izadi, S. (2015). Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones. *IEEE Transactions on Visualization and Computer Graphics, 21*(11), 1251–1258.

Park, C., Moghadam, P., Kim, S., Elfes, A., Fookes, C., & Sridharan, S. (2018). Elastic lidar fusion: Dense map-centric continuous-time slam. *Paper presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA)*.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2*(11), 559–572.

Peretroukhin, V., Wagstaff, B., Giamou, M., & Kelly, J. (2019). Probabilistic regression of rotations using quaternion averaging and a deep multi-headed network. *arXiv preprint arXiv:1904.03182*.

Prasad, V., & Bhowmick, B. (2019). Sfmlearner++: Learning monocular depth & ego-motion using meaningful geometric constraints. *Paper presented at the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Radwan, N., Valada, A., & Burgard, W. (2018). Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters, 3*(4), 4407–4414.

Ramezani, M., Tinchev, G., Iuganov, E., & Fallon, M. (2020). Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure. *arXiv preprint arXiv:2001.10249*.

Rossi, M., Trslić, P., Sivčev, S., Riordan, J., Toal, D., & Dooly, G. (2018). Real-time underwater StereoFusion. *Sensors, 18*(11), 3936.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *Paper presented at the European conference on computer vision*.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *Paper presented at the 2011 International conference on computer vision*.

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.

Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H., & Davison, A. J. (2013). Slam++: Simultaneous localisation and mapping at the level of objects. *Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition*.

Scharstein, D., & Pal, C. (2007). Learning conditional random fields for stereo. *Paper presented at the 2007 IEEE Conference on Computer Vision and Pattern Recognition*.

Schlegel, D., Colosi, M., & Grisetti, G. (2018). Proslam: Graph SLAM from a programmer's perspective. *Paper presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA)*.

Shiguang, W., & Chengdong, W. (2017). An improved FastSLAM2. 0 algorithm using Kullback-Leibler Divergence. *Paper presented at the 2017 4th International Conference on Systems and Informatics (ICSAI)*.

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., & Fitzgibbon, A. (2013). Scene coordinate regression forests for camera relocalization in RGB-D images. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Singh, A. K., & Singla, R. (2020). Different Approaches of Classification of Brain Tumor in MRI Using Gabor Filters for Feature Extraction. In *Soft Computing: Theories and Applications* (pp. 1175–1188). Springer.

Smith, M., Baldwin, I., Churchill, W., Paul, R., & Newman, P. (2009). The new college vision and laser data set. *The International Journal of Robotics Research, 28*(5), 595–599.

Steckel, J., & Peremans, H. (2013). BatSLAM: Simultaneous localization and mapping using biomimetic sonar. *PLoS ONE, 8*(1), e54076.

Steux, B., & El Hamzaoui, O. (2010). tinySLAM: A SLAM algorithm in less than 200 lines C-language program. *Paper presented at the 2010 11th International Conference on Control Automation Robotics & Vision*.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. *Paper presented at the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Sun, Z., Bebis, G., & Miller, R. (2006). Monocular precrash vehicle detection: Features and classifiers. *IEEE transactions on Image Processing, 15*(7), 2019–2034.

Taketomi, T., Uchiyama, H., & Ikeda, S. (2017). Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications, 9*(1), 16.

Tang, S., Zhu, Q., Chen, W., Darwish, W., Wu, B., Hu, H., & Chen, M. (2016). Enhanced RGB-D mapping method for detailed 3D indoor and outdoor modeling. *Sensors, 16*(10), 1589.

Teixeira, B., Silva, H., Matos, A., & Silva, E. (2020). Deep Learning for Underwater Visual Odometry Estimation. *IEEE Access, 8*, 44687–44701.

Tinchev, G., Penate-Sanchez, A., & Fallon, M. (2019). Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a CPU. *IEEE Robotics and Automation Letters, 4*(2), 1327–1334.

Tola, E., Lepetit, V., & Fua, P. (2009). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(5), 815–830.

Turan, M., Almalioglu, Y., Araujo, H., Konukoglu, E., & Sitti, M. (2018). Deep endovo: A recurrent convolutional neural network (rcnn) based visual odometry approach for endoscopic capsule robots. *Neurocomputing, 275*, 1861–1870.

Ullah, I., Su, X., Zhang, X., & Choi, D. (2020). Simultaneous Localization and Mapping Based on Kalman Filter and Extended Kalman Filter. *Wireless Communications and Mobile Computing, 2020*, 2138643. https://doi.org/10.1155/2020/2138643

Valada, A., Radwan, N., & Burgard, W. (2018). Deep auxiliary learning for visual localization and odometry. *Paper presented at the 2018 IEEE international conference on robotics and automation (ICRA)*.

Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., & Fragkiadaki, K. (2017). Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv: 1704.07804*.

Wang, S., Clark, R., Wen, H., & Trigoni, N. (2017). Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. *Paper presented at the 2017 IEEE International Conference on Robotics and Automation (ICRA)*.

Wang, Z., Fan, B., & Wu, F. (2011). Local intensity order pattern for feature description. *Paper presented at the 2011 International Conference on Computer Vision*.

Wang, Z., Peng, Z., Guan, Y., & Wu, L. (2019). Manifold regularization graph structure auto-encoder to detect loop closure for visual SLAM. *IEEE Access, 7*, 59524–59538.

Whelan, T., Leutenegger, S., Salas-Moreno, R., Glocker, B., & Davison, A. (2015). *ElasticFusion: Dense SLAM without a pose graph*.

Woo, A. (2019). *Multi-objective Mapping and Path Planning using Visual SLAM and Object Detection*. University of Waterloo.

Xia, L., Cui, J., Shen, R., Xu, X., Gao, Y., & Li, X. (2020). A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots. *International Journal of Advanced Robotic Systems, 17*(3), 1729881420919185.

Xiao, L., Wang, J., Qiu, X., Rong, Z., & Zou, X. (2019). Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems, 117*, 1–16.

Xu, X., Tian, L., Feng, J., & Zhou, J. (2014). OSRI: A rotationally invariant binary descriptor. *IEEE Transactions on Image Processing, 23*(7), 2983–2995.

Xu, Y., Zhang, S., Li, J., Liu, H., & Zhu, H. (2021). Extracting terrain texture features for landform classification using wavelet decomposition. *ISPRS International Journal of Geo-Information, 10*(10), 658.

Xuexi, Z., Guokun, L., Genping, F., Dongliang, X., & Shiliu, L. (2019). SLAM Algorithm Analysis of Mobile Robot Based on Lidar. *Paper presented at the 2019 Chinese Control Conference (CCC)*.

Yang, X., & Cheng, K.-T. (2012). LDB: An ultra-fast feature for scalable augmented reality on mobile devices. *Paper presented at the 2012 IEEE international symposium on mixed and augmented reality (ISMAR)*.

Yang, J., Li, Y., Cao, L., Jiang, Y., Sun, L., & Xie, Q. (2019). A Survey of SLAM Research based on LiDAR Sensors. *The International Journal of Sensor, 1*(1), 1003.

Yi, K. M., Trulls, E., Lepetit, V., & Fua, P. (2016). Lift: Learned invariant feature transform. *Paper presented at the European Conference on Computer Vision*.

Yin, Z., & Shi, J. (2018). Geonet: Unsupervised learning of dense depth, optical flow and camera pose. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Yousif, K., Bab-Hadiashar, A., & Hoseinnezhad, R. (2015). An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intelligent Industrial Systems, 1*(4), 289–311.

Zaffar, M., Ehsan, S., Stolkin, R., & Maier, K. M. (2018). Sensors, slam and long-term autonomy: A review. *Paper presented at the 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*.

Zhan, H., Garg, R., Saroj Weerasekera, C., Li, K., Agarwal, H., & Reid, I. (2018). Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, X., Su, Y., & Zhu, X. (2017). Loop closure detection for visual SLAM systems using convolutional neural network. *Paper presented at the 2017 23rd International Conference on Automation and Computing (ICAC)*.

Zhang, D., Wong, A., Indrawan, M., & Lu, G. (2000). Content-based image retrieval using Gabor texture features. *IEEE Transactions Pami, 13*.

Zhao, W., He, T., Sani, A. Y. M., & Yao, T. (2019). Review of SLAM Techniques For Autonomous Underwater Vehicles. *Paper presented at the Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*.

Zhao, Y., Wang, T., Qin, W., & Zhang, X. (2018). Improved Rao-Blackwellised particle filter based on randomly weighted particle swarm optimization. *Computers & Electrical Engineering, 71*, 477–484.

Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. *Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhu, A. Z., Liu, W., Wang, Z., Kumar, V., & Daniilidis, K. (2018). Robustness meets deep learning: An end-to-end hybrid pipeline for unsupervised learning of egomotion. *arXiv preprint arXiv:1812.08351*.

Zou, D., & Tan, P. (2012). Coslam: Collaborative visual slam in dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(2), 354–366.