Review Paper

Feature-based visual simultaneous localization and mapping: a survey

Rana Azzam[1] · Tarek Taha[2] · Shoudong Huang[3] · Yahya Zweiri[4]

Received: 30 October 2019 / Accepted: 8 January 2020 / Published online: 16 January 2020

Abstract
Visual simultaneous localization and mapping (SLAM) has attracted high attention over the past few years. In this paper, a comprehensive survey of the state-of-the-art feature-based visual SLAM approaches is presented. The reviewed approaches are classified based on the visual features observed in the environment. Visual features can be seen at different levels; low-level features like points and edges, middle-level features like planes and blobs, and high-level features like semantically labeled objects. One of the most critical research gaps regarding visual SLAM approaches concluded from this study is the lack of generality. Some approaches exhibit a very high level of maturity, in terms of accuracy and efficiency. Yet, they are tailored to very specific environments, like feature-rich and static environments. When operating in different environments, such approaches experience severe degradation in performance. In addition, due to software and hardware limitations, guaranteeing a robust visual SLAM approach is extremely challenging. Although semantics have been heavily exploited in visual SLAM, understanding of the scene by incorporating relationships between features is not yet fully explored. A detailed discussion of such research challenges is provided throughout the paper.

Keywords Robotics · SLAM · Localization · Sensors · Factor graphs · Semantics

1 Introduction

Following several decades of exhaustive research and intensive investigation, Simultaneous Localization and Mapping (SLAM) continues to dominate a magnificent share of the research conducted in the robotics community. SLAM is the problem of concurrently estimating the position of a robotic vehicle navigating in a previously unexplored environment while progressively constructing a map of it. The estimation is done based on measurements collected by means of sensors mounted on the vehicle including: vision, proximity, light, position, and inertial sensors, to name a few. SLAM systems employ these measurements in a multitude of various methods to localize the robot and map its surroundings. However, the building blocks of any SLAM system include a set of common components such as: map/trajectory initialization; data association; and loop closure. Different estimation techniques can then be used to estimate the robot's trajectory and generate a map of the environment.

The implementation details of every SLAM approach relies on the employed sensor(s), and hence on the data collected from the environment. In this paper, we thoroughly review the most recent visual SLAM systems with focus on the feature-based approaches, where conventional vision sensors such as monocular, depth, or stereo cameras are employed to observe the environment. From here on, visual SLAM systems are referred to as monocular SLAM, RGB-D SLAM, or stereo SLAM if they employ a monocular camera, an RGB-D camera, or a stereo camera, respectively.

✉ Rana Azzam, rana.azzam@ku.ac.ae; Tarek Taha, tarek.taha@algorythma.com; Shoudong Huang, Shoudong.Huang@uts.edu.au; Yahya Zweiri, y.zweiri@kingston.ac.uk | [1]Khalifa University of Science and Technology, Abu Dhabi, UAE. [2]Algorythma's Autonomous Aerial Lab, Abu Dhabi, UAE. [3]University of Technology Sydney, Sydney, Australia. [4]Faculty of Science, Engineering and Computing, Kingston University London, Kingston, UK.

SN Applied Sciences (2020) 2:224 | https://doi.org/10.1007/s42452-020-2001-3

The non-conventional event-based vision sensor, such as the asynchronous time based image sensor (ATIS) [98] and the dynamic and active pixel vision sensor (DAVIS) [11], can also be used to solve the SLAM problem as proposed in [64, 131, 132]. Its operation principle is biologically inspired, where instead of capturing frames at a set rate, it asynchronously captures events, which are timestamped changes in brightness of independent pixels. Due to its unique way of acquiring information from the environment, a paradigm shift is necessary to construct algorithms that accommodate such information. Event-based SLAM is beyond the scope of this review paper and interested readers are referred to the comprehensive survey in [41].

Some SLAM systems depend solely on visual measurements, while others augment them with different observations such as range or inertial measurements. Fusion of multiple types of observations might escalate the complexity of the algorithm, require more computational resources, and increase the cost of the platform. However, it makes the system more reliable, robust to outliers, and resilient to failures.

To choose the vision sensor suited for the developed visual SLAM system, the following should be considered. It is not possible to discern the scale of the environment based on observations from a single monocular frame. To compensate for that, monocular SLAM systems adopt different approaches to deduce the depth such as employing a set of one or more other sensors to obtain measurements from which the depth can be deduced, hypothesizing the depth of the observed features using neural networks for example, or by exploiting prior information about the environment, like the size of an observed feature. RGB-D cameras can provide information about depth from a single frame, but they are very sensitive to light, which may limit their applications or the environments in which they can successfully operate. Stereo cameras overcome the limitations of monocular and RGB-D cameras but they are more expensive and resource extensive. The choice of the vision sensor is also dependent on the robotic platform to be used. For instance, ground vehicles do not have any constraints with regards to the weight of on-board sensors, which makes all the options open. However, if an aerial vehicle is to be used, a monocular camera seems to be the most convenient option since it can be seamlessly accommodated on-board, due to its lightweight, small size, and low power requirements. Nevertheless, the employed algorithms must deal with the scale ambiguity of the obtained visual observations.

Visual measurements can be handled at different levels of detail. Direct SLAM systems, for example: [34, 85, 86], process the intensities of all or a subset of pixels in the image. Then, based on the brightness consistency constraint [139], correspondences are established between multiple observations. Feature-based SLAM, on the other hand, targets features that exhibit distinctive properties and can be repeatedly detected by the employed detection algorithms. Examples of such systems include [65, 91, 97]. Features can be classified into different levels; low-level features such as points, corners, and lines, medium-level features such as blobs and planes, and high-level features such as objects as illustrated in Fig. 1. A visual SLAM system might employ a single [23, 45, 88] or a hybrid [10, 54, 138] of different feature levels.

In our review, we classify the state-of-the-art feature-based visual SLAM solutions based on the features used to perform localization and mapping. Within each category, implementation choices of the adopted SLAM pipeline are thoroughly discussed and compared. Strengths and weaknesses of each category are highlighted and open research problems are emphasized at the end.

1.1 Existing surveys on SLAM

The proposed approaches to SLAM were surveyed by several researchers in the field and the open research problems to-date were highlighted. In [14], the authors argued that SLAM is entering the robust perception era

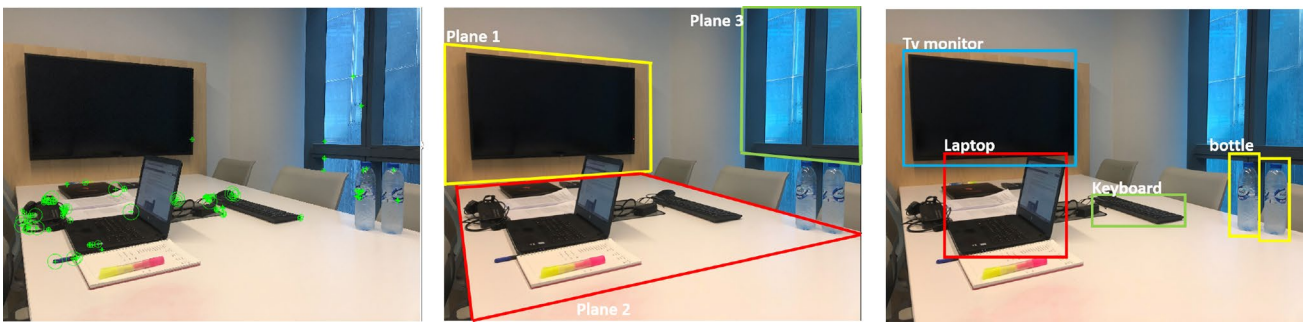


Fig. 1 Different visual features extracted from the same visual frame. Left: low-level features (SURF [6]), middle: middle-level features (planes), right: high-level features (semantically labeled objects)

and thoroughly discussed the main characteristics of state-of-the-art solutions in terms of several performance metrics such as scalability, robustness, and representation. In addition, the paper addressed the recent advancements at the hardware and algorithmic levels and pointed out the research problems that are yet to be solved. A comprehensive review of key-frame based approaches to SLAM was presented in [139] where the general architecture of key-frame based monocular SLAM and the corresponding implementation approaches were presented. The survey conducted in [109] targeted SLAM approaches that omit the assumption that the environment under investigation is static and addressed the underlying techniques adopted to reconstruct a dynamic environment. Along the same lines, the survey presented in [96] studied the SLAM approaches that can operate in dynamic environments and those that employ heterogeneous data that can be obtained through a visual sensor, for instance: color, depth, and semantic information. Visual SLAM approaches that rely on observing primitive features in the scene were surveyed in [44] and classified according to the descriptors used for such features, emphasizing their strengths and weaknesses. An overview of the anatomy of visual odometry and visual SLAM, along with the underlying formulations and implementation choices was provided in [141]. Similarly, in [40], the solutions to visual SLAM were analyzed based on their implementation of the main building blocks of SLAM, and their failure in dynamic environments was analyzed. The SLAM approaches reviewed in [120] were classified into feature-based approaches, direct approaches, and RGB-D based approaches. Comparisons between the state-of-the-art solutions back in 2016 were conducted, followed by a set of open research problems relating to the mentioned categories. Finally, a recent survey on SLAM, with focus on semantics can be found in [115]. In this paper, we contribute a comprehensive survey of the most recent state-of-the-art feature-based visual SLAM systems and we classify the reviewed approaches based on the elements, i.e. features, they extract from visual frames to localize the robot and reconstruct the environment. Such features fall in one of the following categories: low-level, middle-level, or high-level features. So, the reviewed approaches are classified as shown in Fig. 2. Our review serves as a thorough reference for researchers interested in investigating the various implementation options and advances in feature-based visual SLAM. Approaches that fall into the same feature-level category were further grouped based on other goals that they accomplish, like real-time performance, handling scene dynamics, and resilience to data association failures. The techniques that made each of these goals possible were listed and analyzed. This will assist the readers to accurately determine what makes out each of these approaches and what

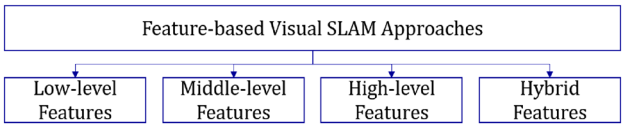


Fig. 2 Classification of feature-based visual SLAM approaches

implementation methods they need to adopt and/or improve to develop a system that can achieve a particular set of objectives.

The rest of this paper is organized as follows. The anatomy of a generic SLAM system is presented in Sect. 2 where the SLAM building blocks are discussed in detail along with different implementation options. The review and analysis of the feature-based visual SLAM systems and their design choices are provided in Sect. 3. In Sect. 4 we highlight the outcomes of our review and identify the open problems that need further investigation.

2 SLAM building blocks

Before delving into the implementation details of the current state-of-the-art solutions, the common components of visual SLAM are briefly discussed, including (1) Map/Trajectory Initialization, (2) Data Association, (3) Loop Closure, (4) Relocation, and (5) Estimation Algorithms, as shown in Fig. 3. The purpose of each component is first provided, followed by the most prevalent implementation approaches, when applicable.

2.1 Map/trajectory initialization

Upon starting a robotic task in a new environment, a map of which is not available a priori, it is necessary to estimate the 3D structure of the surroundings as well as the position of the robot with respect to it. This serves as an initial assessment of the map that will be iteratively updated based on the sensory measurements collected throughout the task. This process is only required to bootstrap the system at startup. There are several ways in which initialization can be carried out when different sensors are employed. For instance, one depth frame or a stereo pair are sufficient to initialize a map, as presented in [97, 118], since they provide depth and scale information, which monocular frames lack. On the other hand, initialization can be done manually when monocular cameras are in operation, for example [32], where the system is provided with prior information about the observed scene, which include the positions and appearance of four features, resolving the scale ambiguity problem. Examples of other algorithms that are commonly used for map initialization

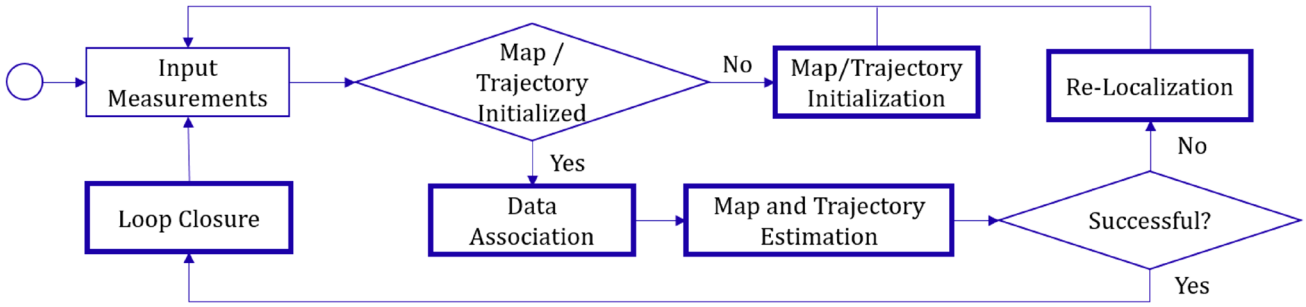


Fig. 3 SLAM Pipeline

are iterative closest point (ICP) [8, 20], image alignment [80, 112], five-point algorithm [114] together with a model fitting algorithm such as random sample consensus (RANSAC) [38] or MLESAC [123], and inverse depth parameterization relative to the camera, which is used to parameterize observed features [25]. Kinematic models, for example [22], and integration of inertial measurements, as presented in [29], can be used to initialize the trajectory.

2.2 Data association

While maneuvering in the environment, the robot may sense the same area multiple times. Establishing correspondences between the image frames, collected each time the same scene was observed, is of paramount significance to estimate the map and the robot's trajectory, and is referred to as data association.

Feature-based approaches target *features*, which are areas in the image that exhibit distinctive properties. Features can be of different scales; low-level features such as geometric primitives, middle-level features such as super-pixels, or high-level features such as semantically labeled objects. The most critical characteristic of a feature is *repeatability*, which makes the feature detectable repeatedly when appearing in multiple frames taken from different viewpoints.

To detect features in an image, several detectors were proposed in the literature for different feature types. For low-level features, such as points, lines, edges, and corners, Table 1 shows some examples of feature detectors as well as descriptors. After detecting a feature, it is extracted from the image together with its surrounding pixels, then assigned a quantitative measure, referred to as a descriptor, to facilitate matching with other features.

To detect planes in images, model fitting algorithms, such as RANSAC, are employed. It is also possible to combine modeling and a convolutional neural network (CNN) to identify planes, such as walls, in an image [136]. As for high-level features, several techniques were proposed for detecting objects and semantically labeling them in images including, but not limited to, conditional random fields (CRFs) [51], support vector machines (SVMs) [30], and deep neural networks (for example: single shot multi-box detector [74] and you only look once (YOLO) [104]).

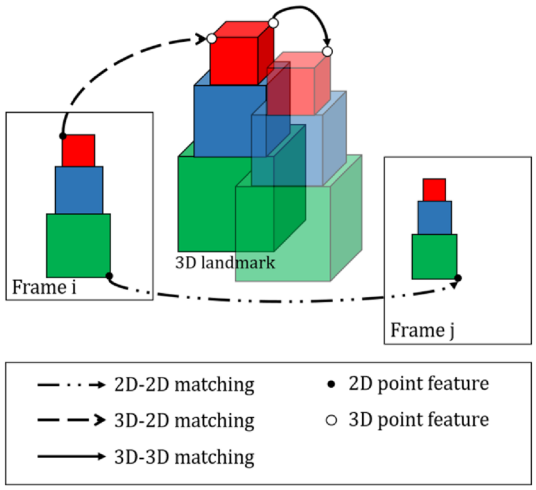
Table 1 Feature detectors and descriptors

Feature detectors	Feature descriptors
Hessian corner detector [7]	BRIEF [16]
Harris detector [49]	SURF [6]
Shi–Tomasi corners [111]	SIFT [76]
Laplacian of Gaussian detector [72]	HoG [31]
MSER [83]	ORB [107]
Difference of Gaussian [77]	FREAK [1]
FAST/AGAST/OAST [81]	BRISK [68]

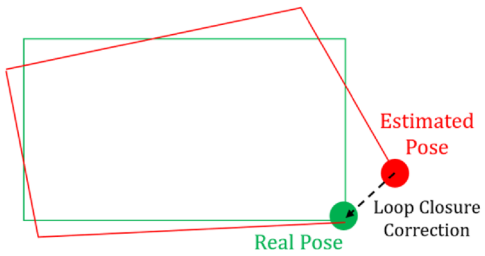
Establishing correspondences between low-level features can be done between features in two images (2D-2D matching), between a point in the 3D map and its projection onto the image frame (3D-2D matching), or between two 3D points in the reconstructed map (3D-3D matching) [140] as depicted in Fig. 4a.

Matching a feature in the current image to a feature in another image (2D-2D) is performed by means of a search within a window in the second image enclosing the location of the feature in the current image. The search is reduced to one dimension if the transformation between both images is known and hence, epipolar geometry [50] can be established. The similarity between the features' descriptors can be measured using different quantities depending on their types, such as the sum of squared distance, L1/L2 Norms, or the hamming distance, to name a few. Such measures might hinder the performance of the system due to their high computational requirements and can be replaced by kd-tree search, similar to [89], or bag of binary words approaches such as [42].

3D-2D matching is necessary when the pose of the camera needs to be estimated given the 3D structure of the environment. 3D points surrounding a hypothesized pose are projected onto the current image frame. 2D



(a) Data association for low-level features



(b) Loop closure example

Fig. 4 Data association and loop closure examples

projections are then matched to 2D features in the image using the previously mentioned techniques.

Upon re-visiting a location, i.e. closing a loop, the corresponding 3D landmarks are matched (3D-3D) yielding a corrected, drift-free path.

Establishing associations between middle-level features, such as planes, is done by comparing plane parameters, such as normals (for example: [54]), the overlap, and the distance between the plane detected in the current frame and those available in the map (such as [138]). If the distance is below a particular threshold, correspondences are established. Otherwise, a new plane is added to the map.

In order to establish correspondences between semantically labeled landmarks, the predicted label is used to associate a detection with a landmark in the map. In case multiple instances of the same object category appear in the environment, a minimum distance threshold between them must be exceeded to consider inserting a new landmark into the map [10]. Otherwise, the detection is associated with its closest landmark. In a recently proposed SLAM solution [95], objects are detected and characterized at the category level rather than just the instance level. This is based on the fact that all objects in one class have common 3D points irrespective of their different categories. Other approaches to data association will be discussed in more detail in the next section.

2.3 Loop closure

As the robot progresses through its task, errors from several sources accumulate causing the estimation to drift off the real trajectory (An example is illustrated in Fig. 4b). Such drift may severely affect the reconstruction of the environment and hence lead to failure of the ongoing robotic task. To correct such drift, several techniques were proposed in the literature to detect loop closure, i.e. to detect whether or not the currently observed scene was assessed by the robot earlier, and hence achieve global consistency. Global consistency is the condition where the SLAM estimate matches, approximately, the ground truth and the reconstructed map conforms to the real topological structure of the observed area. However, local consistency refers to the case where the observations are matched locally but, perhaps, not globally [84].

Loop closures usually involve two main steps: visual place recognition and geometric verification. The former can be done using kd-tree search [75], bag of words approaches [63], Bayesian filtering [2], deep learning [43, 135], and visual feature matching [53, 79], while the latter can be achieved through image alignment, and RANSAC [38].

2.4 Re-localization

Re-localization is the ability of a SLAM system to recover from a fatal localization failure in which the robot is assigned an arbitrary location. This failure can result due to several reasons, such as abrupt motions, motion blur, or absence of features [139]. Moreover, the robotic vehicle might be re-positioned through an operation that is out of the robot's control, in which case the robot's global position is to be determined [12]. These cases are referred to as the Kidnapped Robot problem [35] and can be resolved using several techniques, including but not limited to, matching feature descriptors [71], re-observing semantically labelled objects [48, 106], epipolar geometry [82], or bags of binary words approach [91, 105],

2.5 Estimation algorithms

Estimation algorithms are needed to resolve the SLAM constraints, and can be classified into batch and incremental algorithms. Batch algorithms, such as global bundle adjustment (GBA) [125] and full graph SLAM [122], process a large set of measurements collected by the robot, over

a relatively large period of time, to reconstruct the map of the environment as well as the robot's trajectory. Incremental algorithms, on the other hand, compute estimates of the map and trajectory upon arrival of new measurements. Some incremental algorithms, such as [61] operate on the entire set of measurements collected throughout the robotic task, while others, such as [60] operate on a subset of those measurements collected over a small time frame, which facilitates operation in an online manner. While batch algorithms succeed in achieving global consistency, they are computationally expensive, and hence, may impede real-time operation. In addition, due to the constrained memory resources, they might not work for large-scale environments or for continuously operating systems, which emphasizes the significance of incremental algorithms that do not suffer from such limitations. Revisiting old data association decisions is not possible when estimation is done through incremental algorithms that do not consider all measurements, which may increase the cumulative error compared to other algorithms. In what follows, batch algorithms, such as GraphSLAM [122] and GBA [125], as well as incremental algorithms, such as extended Kalman filter (EKF) [122], incremental smoothing and mapping [60, 61], and local bundle adjustment (LBA) [87], are briefly presented.

2.5.1 Extended Kalman filter (EKF) [122]

Given multiple measurements recorded over a period of time, possibly from several sensors, an EKF estimates the state of the system under observation. The state of a system consists of the states of both the environment and the robotic vehicle. The former describes the poses of the landmarks observed in the environment, while the latter describes the vehicle's kinematics. The estimation process involves filtering the noise associated with each measurement to reduce the overall uncertainty of the estimated state. Then, EKF estimates the states of the system through several iterations of predictions and updates based on the measurements collected from the environment, as depicted in Fig. 5.

2.5.2 Factor graph SLAM [122]

As the name of this algorithm suggests, a graph is used to reconstruct the map of an environment along with the robot's trajectory in it. Map features and robot poses are represented as vertices, and are connected using edges that encode two types of nonlinear constraints: motion and measurements as shown in Fig. 6a. The summation of all the constraints makes SLAM a nonlinear least squares problem. To obtain an estimate that is globally consistent, all constraints are first linearized, yielding a sparse

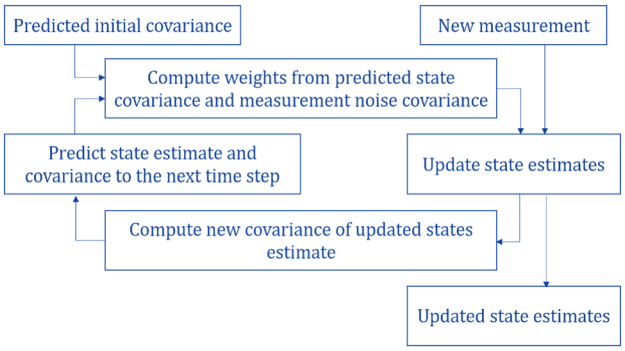


Fig. 5 The extended Kalman filter algorithm [108]

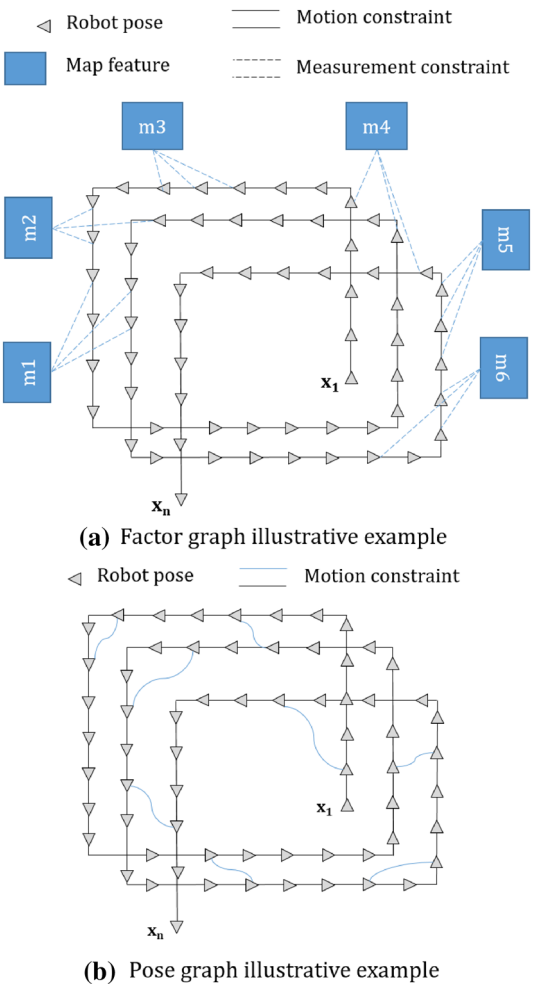


Fig. 6 Factor and pose graph examples

information matrix and an information vector. Due to the sparseness of the matrix and for a more efficient computation, the matrix is reduced in size using a variable elimination algorithm. An inference technique is then employed to find the assignment of poses to the nodes of the graph which minimizes the errors imposed by the constraints.

Alternatively, successive robot poses in the environment could be used alone to estimate the location of the robot using a Pose Graph [119]. The graph used in this problem includes the robot poses as nodes and motion constraints as edges between those nodes as depicted in Fig. 6b.

Bundle adjustment (BA) [125] is an instance of factor graph SLAM and can be defined as a refining process that simultaneously optimizes the 3D structure, the camera trajectory, and possibly its calibration parameters using a sequence of images collected from the environment as depicted in Fig. 7. A cost function that assesses the error in the system is minimized to yield an improved estimation of the reconstruction. If all measurements since the beginning of the robotic task were considered in the estimation, the process is referred to as GBA, and is known to be computationally expensive which hinders online operation [36].

A more computationally efficient approach that incrementally adjusts the 3D reconstruction and camera trajectory was proposed in [87] and is referred to as Local Bundle Adjustment (LBA). Only a window of *n* recent frames is adjusted upon reception of a new measurement. Using LBA makes it possible to execute SLAM in real time.

ParallaxBA is another variation of BA that was presented in [145] where features are parameterized using parallax angles instead of their Euclidean coordinates or inverse depth, ParallaxBA outperformed traditional BA in terms of accuracy and convergence.

2.5.3 Incremental smoothing and mapping

Incremental Smoothing and Mapping is an approach to SLAM that gradually computes estimations of the map and robot trajectory while measurements are being collected from the environment. Several approaches were proposed in the literature, the most prevalent of which are iSAM [61] and iSAM2 [60]. iSAM performs smoothing using QR factorization of the square root information matrix and iSAM2 operates on a novel data structure referred to as the Bayes tree which is obtained from factor graphs.

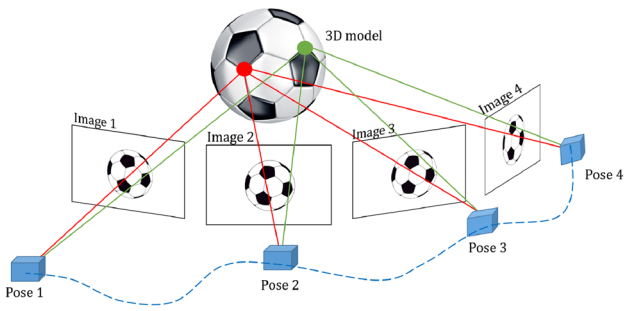


Fig. 7 Bundle adjustment illustrative example

3 Feature-based visual SLAM—design choices

In this section, an overview of the state-of-the-art feature-based visual SLAM systems is presented. As mentioned earlier, features can be of different granularities; low-level features, middle-level features, or high-level features. A visual SLAM system may be based on the use of either one or a hybrid of two or more feature types as will be discussed in the following sections. The most alarming concern about feature-based approaches is their failure in absence of features. Regardless of their high achievable performance and accuracy in feature-rich environments, if the environment under investigation lacks the features that visual SLAM relies on, be it points, planes, or objects, localization fails and the estimation of the robot's surroundings would not reflect the true structure. In what follows, visual SLAM systems are classified and discussed according to the types of features employed in the system.

3.1 Low-level feature-based approaches

Low-level features are geometric primitives that are observable in abundance in textured scenes. The vast majority of the existing visual SLAM systems, for instance [22, 62, 70, 91, 127] exploit these features, throughout the localization and mapping processes, and have achieved a very high-level of maturity in terms of accuracy and efficiency. However, if the environment in which the robot is operating is texture-less or lacks the features that the system can track, such methods fail due to the absence of features, hence why the most recent SLAM approaches started to consider features at different levels at the same time.

3.1.1 Multiple feature types to aid robustness

Feature-based visual SLAM systems that depend on a single type of features are susceptible to failure when such features do not exist in the environment under investigation. To circumvent this issue, the work presented in [99] proposes using points and lines together to perform monocular SLAM in a poorly textured environment. Lines are parameterized by their endpoints to facilitate integration with point-based approaches. In absence of point features, this work proposes a novel technique to initialize the system using lines only. The same set of landmarks were also adopted to perform stereo SLAM in [47]. Stereo visual odometry is used to track points and lines, and Gauss

Newton optimization is then employed to estimate the motion of the camera by minimizing the projection errors of the corresponding features. In [59], observations of point features are combined with laser scans and used in a factor graph to estimate the pose of the robot. A new map representation combining both an occupancy grid map and point features was proposed. By matching observed features to landmarks in the map, loop closure and localization can be achieved efficiently. Hence, the flexibility regarding what type of feature to adopt while estimating the robot's trajectory in the environment greatly benefits the robustness of visual SLAM.

3.1.2 Facilitating real-time performance

The maps generated by low-level features are sparse yet require large computational and memory resources. This is attributed to the fact that the process of detecting, extracting, and matching features is one of the most computationally expensive blocks in the SLAM pipeline.

In order to achieve real-time performance, some systems [27, 94, 97, 134] heavily exploit parallelism to perform tracking and mapping as originally proposed in PTAM [65]. Two threads are concurrently run to localize the robot and map its surroundings [65, 97]. Unlike tracking, delays are tolerable in the mapping thread where most of the heavy computations take place. To further reduce computations, [94] limited the number of features to be extracted, and used a local map through which feature matching is performed. In order to maximize parallelism, a separate thread was employed to perform loop closing and a synchronization process was proposed where access to map points is granted to a thread only if the points are not currently being processed by another thread.

In [27], three parallel modules are employed; *scene flow* for feature detection, extraction, and matching, *visual odometry* for camera motion estimation, and *global SLAM* for loop closing and global consistency.

Localization and mapping can also be done in a distributed manner by multiple robotic vehicles while exploiting parallelism as proposed in [134] where tracking and image acquisition, which are lightweight processes, are run on-board all MAVs in parallel while mapping is done off-board by a powerful computer due to its computational demands. A recent monocular SLAM system was proposed in [102] where EKF and BA were exploited together to achieve real-time robust performance. ORB features and inertial measurements were used in a visual inertial odometry (VIO) framework based on EKF which is capable of estimating the camera motion with minimal delays. To further assist real-time performance, not all ORB features are extracted from visual frames in the VIO framework which operates on all incoming frames. In addition, to circumvent the estimation errors resulting from EKF, a globally consistent map estimated using BA is frequently updated based on selected keyframes and fed to EKF to correct any estimation errors. The selected keyframes go through another round of feature extraction and matching since the features extracted for VIO are not sufficient for building a robust map. Loop closure is run in a parallel thread to correct accumulated error by performing place recognition and ORB feature matching. Once a loop is detected, pose graph optimization as well as GBA are carried out. Due to the fusion of visual and inertial measurements, the approach is robust to abrupt motions and is capable of resolving scale ambiguity. It also combines the advantages of EKF and BA to achieve real-time performance and robustness respectively.

Figure 8 summarizes the techniques that can be adopted to speed up the localization and mapping processes and get the estimation done in real-time.

3.1.3 Resolving scale ambiguity

When using a monocular camera, a SLAM system needs to handle the inherent scale ambiguity challenge which results from the difficulty to discern depth from a single frame. An EKF based approach was proposed in [127] where scale ambiguity and intermittent absence of features are compensated for by fusion of monocular vision, ultrasonic and atmospheric pressure measurements. Fusion of multiple sensors was also seen in [78] where vision, inertial, and range measurements were employed to achieve the objectives of SLAM. Scale ambiguity in [82] is circumvented by two-view initialization. A pair of images is selected according to their relative rotation, Euclidean

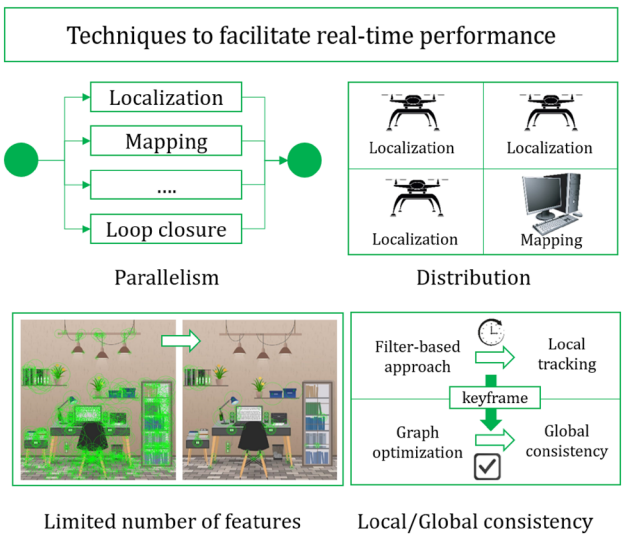


Fig. 8 Techniques to facilitate real-time performance

distance, and the time difference between them. Then, epipolar geometry is used to estimate the scale based on the matched features between these frames. In another monocular SLAM approach [142], the depth of ORB features was computed based on their distance to the vanishing points identified in the scene. Furthermore, inverse depth parameterization was used in [26] to recover the scale of the scene.

While not required for RGB-D and stereo SLAM, adopting a technique to resolve the scale of the map is essential for monocular SLAM. Figure 9 illustrates the techniques that can be used to resolve scale ambiguity.

3.1.4 Resilience to feature detection/association failure

Failure to observe or match low-level features in an environment is equivalent to operating in texture-less environments in which feature-based visual SLAM systems fail. In both cases, the system suffers from absence of measurement constraints, causing severe performance degradation. A vision system fails to detect or match features between frames in case of abrupt sensor motions or in presence of dynamics in the scene.

One of the limitations of the original EKF-SLAM, which is described in [122], is its inability to handle abrupt motions. To overcome this, the approach proposed in [73] employs visual input in both phases of the filter; prediction and update. Optical flow and epipolar geometry are used to estimate the state transition of the camera. Using images in the prediction stage made the system robust against abrupt motions and infrequent data acquisition. This has also eliminated the need for dynamic models and resulted in a faster and more efficient performance. Although this EKF variant improves the robustness and efficiency of SLAM in particular cases, it still fails if there are no features in the scene. Another variation of EKF-SLAM is proposed in [100] IMU measurements are used in the prediction phase and RGB-D images are used in the update phase. To achieve global consistency, pose graph optimization is performed. Fusion of IMU measurements made it possible for the system to successfully operate in texture-less and dynamic environments.

ORB-SLAM2 [91] is a state-of-the-art visual SLAM system that performs tracking, mapping, and loop closing based solely on ORB features in real time while running on standard CPUs. Due to its dependence on visual features, ORB-SLAM2 fails in absence of ORB features in the scene. To this end, a tightly-coupled fusion of odometry and ORB-SLAM2 was proposed in [15] where the motion model is replaced by odometry, which supports the estimation when no features can be detected in the scene.

Similarly, the approach proposed in [62], exploits tightly-coupled fusion of inertial and visual measurements to perform visual inertial odometry. Global consistency is then achieved by means of loop closure detection and global pose graph optimization. Another variation of ORB-SLAM2 can be found in [121], where ORB features were replaced by learnt point features, referred to as GCNv2. It was demonstrated that the proposed approach has comparable performance to ORB-SLAM2 in most scenarios, but performs slightly better in case of fast rotations.

Failure to associate features across subsequent frames can also result from dynamics in the scene. The work proposed in [128] demonstrates the ability to successfully perform RGB-D SLAM in a dynamic environment while observing low level features only. Using the fundamental matrix, feature points belonging to moving parts of the scene are extracted. Then, efficient PnP was used to estimate the pose of the camera in the environment. The re-projection errors are then further optimized by means of BA. The proposed approach was successfully used in real experiments but only under the assumptions that there is small parallax and more than 24 point matches between consecutive frames. Hence, the approach fails

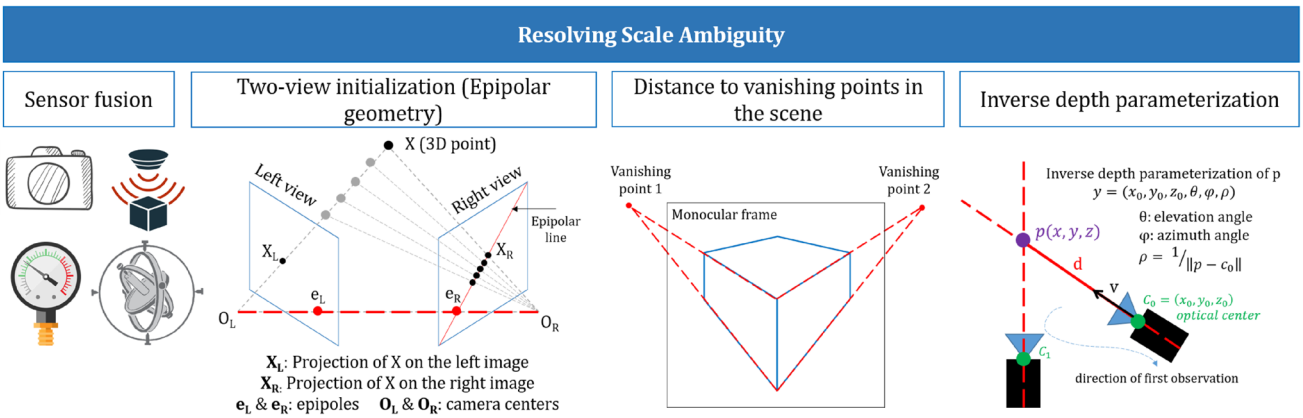


Fig. 9 Techniques to resolve scale ambiguity

to work in presence of abrupt motions and in absence of low-level features in the environment.

In order to enhance the performance of visual SLAM in dynamic environments, the approach proposed in [21] employs a sparse motion removal scheme. A Bayesian filter is used to compute the similarities and differences between consecutive frames to determine the dynamic features. After eliminating such features, the scene is fed to a classical visual SLAM approach to perform pose estimation. This approach works only in presence of features in the scene and fails otherwise.

Another low-level feature based visual SLAM approach that is robust to false data association occurring in dynamic scenes is found in [13]. The approach is based on a novel filter where poses are encoded as dual quaternions. Association of ORB feature observations and map landmarks is done through an optical flow-based approach which makes it robust to dynamics in the scene.

In summary, lack of features, abrupt camera motions, and dynamics in the observed scene are the main reasons behind failure to perform data association. Some techniques that are adopted in the literature to solve these issues include employing multiple sensors that observe different information in the scene and eliminating observations that involve dynamics.

The implementation details of the reviewed low-level feature-based approaches are provided in Table 2.

3.2 Middle-level feature-based approaches

Middle-level features are planes or blobs that are observed in the environment. Using such features as landmarks improves the SLAM performance in texture-less environments where it is challenging to observe low-level features; in corridors for example. To observe such features, model fitting approaches are employed. Hence, there is a trade-off between the estimation accuracy and the time

Table 2 Implementation choices adopted by low-level feature-based approaches

References	SLAM component				
	Initialization	Data association	Loop closure	Estimation	Relocation
[78]	IMU measurements	Sweep matching & approximate nearest neighbor	LSO	Algorithm in [9]	–
[127]	Range measurements	Visual measurement models	–	EKF	–
[22]	Kinematic model	Geometric matching then ICP	–	EKF	–
[62]	–	FM	BBW	FG	BBW
[70]	SI	FM	VT	RANSAC & non-linear refinement	–
[82]	SI	Optical flow	IMI	GBA	EG & BA
[99]	Trifocal tensor eq.	Relational graph strategy	Essential graph	FG and GBA	EPnPL
[73]	IDP	FM	–	EKF	–
[65]	SI & RANSAC	FM	Mapping module	GBA	–
[27]	–	VT and EG	VT	GBA	VT
[101]	PM	Edge alignment using GN	–	GN Optimization + EKF	–
[97]	SI	FM	BA	LM algorithm	–
[94]	SI	FM	PG	Co-visibility graph	–
[91]	FF	2D-3D points matching	BBW	FG and GBA	BBW
[15]	FF	2D-3D points matching	BBW	FG and GBA	Odom
[134]	SI + RANSAC	FM	GBA	GBA	SBI [66]
[100]	FF	FM	FM	FG	–
[3]	Odom	FM	BBW	FG	–
[129]	PM	Multi-hypotheses via PF	–	PF	–
[126]	PM	Distance function	–	EKF	–
[128]	FF	FM	FM	Effiicnet PnP and BA	–
[102]	VIO EKF	ORB FM	PG & GBA	EKF & BA	–

FG factor graph, *PG* pose graph, *BA* bundle adjustment, *EKF* extended Kalman filter, *MAP* maximum-a-posteriori, *VT* vocabulary tree, *PF* particle filter, *FM* feature matching, *BBW* bags of binary words, *GN* Gauss Newton, *LM* Levenberg Marquardt, *FF* first frame, *PM* prior map, *SI* stereo initialization, *EG* epipolar geometry, *IDP* inverse depth parameterization, *Odom* Odometry, *SBI* small blurry image relocation, *LSO* least square optimization, *IMI* image moment invariants

– indicates that implementation details about the corresponding element/block are not provided

needed to compute accurate models from the environment. Using those features alone is not common since fusing them with low- and high-level features results in better accuracy as discussed in Sect. 3.4. In [113], a SLAM approach based solely on RGB-D data is proposed. A 3D map of the environment is constructed using planes representing walls and floors while removing all other objects from the scene. RANSAC is employed to estimate planar surfaces which are then refined by estimating their normals and extracting the corresponding convex. Then, an l_0 norm minimization algorithm is used to maintain planes that are highly likely to represent walls or floors while minimizing the inclusion of smaller ones. Using this approach, it was possible to reconstruct a map of the walls and floor as illustrated in Fig. 10. However, no other features are present in the map which makes it unusable for the majority of SLAM application. This motivates the need for considering high-level features, as presented in the next section.

3.3 High-level feature-based approaches

Perceiving high-level features is paramount when robots are expected to perform tasks that require scene understanding such as searching for a victim after a catastrophe, building meaningful maps, and grasping or operating on particular objects in the environment. This is very challenging to achieve with maps reconstructed using low-level features since they lack expressive representation which makes it harder for humans to understand [39, 46]. High-level features add critical information about the structure of the scene and convey the semantic meaning of every part of the reconstructed map. They are environment-specific and may vary in size, shape, and dynamicity. In a city-scale application, possible landmarks include trees, buildings, streets, or sidewalks. On the other hand, furniture, office supplies, and home appliances may serve as landmarks for indoor applications. In this section, different approaches to data association in high-level feature-based SLAM approaches will be thoroughly discussed. Then, techniques to achieve real-time performance and handle dynamics in the scenes will be presented.

3.3.1 Associating high-level feature observations with landmarks

Although high-level features are detected and semantically annotated, data association in the event that multiple instances of the same object category exist in the environment poses a fundamental challenge in high-level feature-based visual SLAM systems [88].

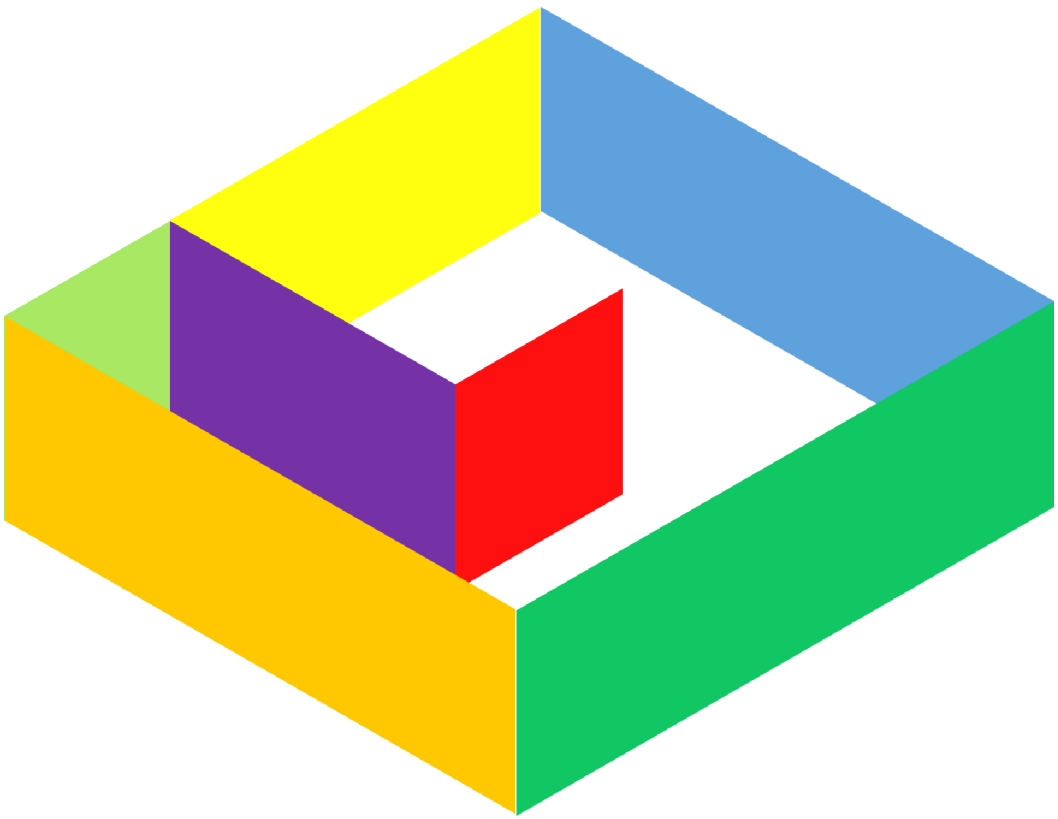


Fig. 10 Illustration of reconstructed map based on planar features

In [95], objects are detected and characterized at the category level rather than just the instance level. This is based on the fact that all objects in one class have common 3D points, irrespective of their different categories. Such points, referred to as keypoints, are used to distinguish between the different categories of the same class. Input monocular frames are passed to an object detector, YOLO9000 [103], and 3D keypoints in the resulting bounding boxes are localized by means of another convolutional neural network. Shapes and poses are optimized using the Ceres solver. Instead of performing object and keypoint detection on every frame, objects are tracked in successive frames leading to higher efficiency and speed.

Another novel data association approach was presented in [45] to localize a robot in a prior map. First, a query graph is computed for each image where a vertex represents an object's class and centroid, and undirected edges between vertices indicate the fulfillment of a proximity requirement. A merged graph for all the images is then created by connecting vertices from consecutive images using the Euclidean distances between them. Vertices that are too close to each other are merged to avoid duplicates. The second step is the generation of a random walk descriptor for each vertex. That is, an $n \times m$ matrix containing the labels of m visited vertices in n random walks. Third, the query graph is to be matched to the global database graph based on a similarity score. The similarity score of two vertices indicates the number of identical rows in their descriptors. The highest k matches are then used to localize the query graph in the database graph.

In [69], semantically labeled objects as well as their inter-relationships are employed in the process of establishing correspondences between input monocular frames. An RGB frame is first passed to a Faster R-CNN to detect objects. Then, the transformation between consecutive images is computed by first generating multiple cuboids that lie along the line, formed by the camera center and the center of the bounding box, and projecting them onto the detected bounding box. Generation of cuboids is done at discrete distances and angles. After that, coordinate descent is performed to minimize the difference between the corners of the detected bounding box and the projection of each cuboid into the image plane. Redundant cuboids are then removed. Each of the remaining cuboids is then used as a seed to generate a scene, which is a set of cuboids each corresponding to a detected bounding box, based on contextual constraints. To find correspondences between the generated set of scenes, a sampling-based approach is used. Every pair of scenes is searched for correspondences based on semantic labels. Three correspondences from every pair are picked and frames of reference for each scene are constructed. The transformation between the scenes is computed accordingly and scored based on how well the remaining correspondences fit using the computed transformation. The sample with the highest rank is then used to estimate the transformation between camera poses.

In [24], an object hypothesis is generated if the same object segment is observed in multiple frames and is represented using 3D feature descriptors which facilitate loop closure. Inlier correspondences between the current object and the objects in the map are computed, then, the object is associated with the hypothesis with which it achieved the highest number of correspondences. If the number of correspondences falls below a threshold, a new object representation is added. Only one or a few static instances of an object category are assumed to be in the environment. A prior estimation of the robot pose based on odometry and ICP is computed using OmniMapper [124]. Based on that, the current frame's segments are projected into a common frame of reference with all previously segmented objects. The centroid of each segment is matched to the closest segment centroid in the map. To verify the match, the bounding box of the current segment and that of the segment to which it was matched are compared. If there is not enough overlap between the bounding boxes, a new object is initialized. The final object model is created by aggregating all the corresponding segments after transforming them according to the relative camera poses. Spatial constraints between the object model and the robot poses are then added to the SLAM system.

In [88], SLAM and data association are addressed as tightly coupled problems and a novel approach is proposed to simultaneously estimate a robot's position and associate its observations with landmarks. A back-end approach was used to jointly solve the object detection and SLAM problems. After being detected, an object is represented by the centroid of its point cloud obtained from RGB-D data. Neither data association nor the total number of landmarks in the environment are known a priori. A probabilistic model based on the Dirichlet Process was hence introduced to establish proper data association. Overall, a mixed-integer nonlinear problem is set up to estimate the robot poses, landmark locations, and data association given the robot's relative poses and observations.

The most common approach to data association in presence of multiple instances of the same object category is the distance threshold as presented in [23]. Each robot in the proposed distributed SLAM framework performs SLAM through OmniMapper [124] based on visual and odometry measurements. Each input RGB image is passed to a YOLO object detector. Detected objects are segmented and the PFHRGB features in their point cloud and in the

corresponding model are extracted and matched. If at least 12 correspondences were detected, generalized iterative closest point (GICP) [110] is performed to compute a refined pose of the object. Data association is then performed by searching for instances of the same detected object category within a distance threshold. Figure 11 summarizes the main approaches found in the literature to perform data association of high-level features.

3.3.2 Facilitating real-time performance

Performing real-time localization and mapping is very critical for some robotic tasks, especially those performed in harsh environment for search and rescue purposes. However, the processing time for some blocks in the pipeline, such as object detection and segmentation, goes beyond that. In this section, focus will be devoted to the techniques used to facilitate real-time performance in high-level feature-based SLAM approaches.

The work proposed in [95] proposes not performing object detection on all the incoming frames. Rather, after detecting an object in a keyframe, it is tracked in successive frames, which significantly reduces the time needed to process the data.

For the same purpose, the system proposed in [24] preprocesses the scene by dividing it into planar and non-planar (object) segments. After removing planar segments, object segments are refined and associated to already existing landmarks in the map.

Representing objects using quadrics is an alternative technique to reduce computations while employing semantically labeled landmarks in a visual SLAM system. The work proposed in [93] uses an object detector as a sensor where the detected bounding boxes are used to identify the parameters of the quadric representing the corresponding object. A quadric provides information about the size of the object, its position, and its orientation, encoded in ten independent parameters. A geometric error formulation was proposed to account for the spatial uncertainty of object detections, resulting from occlusions for example. Using quadrics instead of detailed object models enhances the speed of the system at the expense of reconstructing information-rich maps which are useful in a wide range of applications. An illustration of the

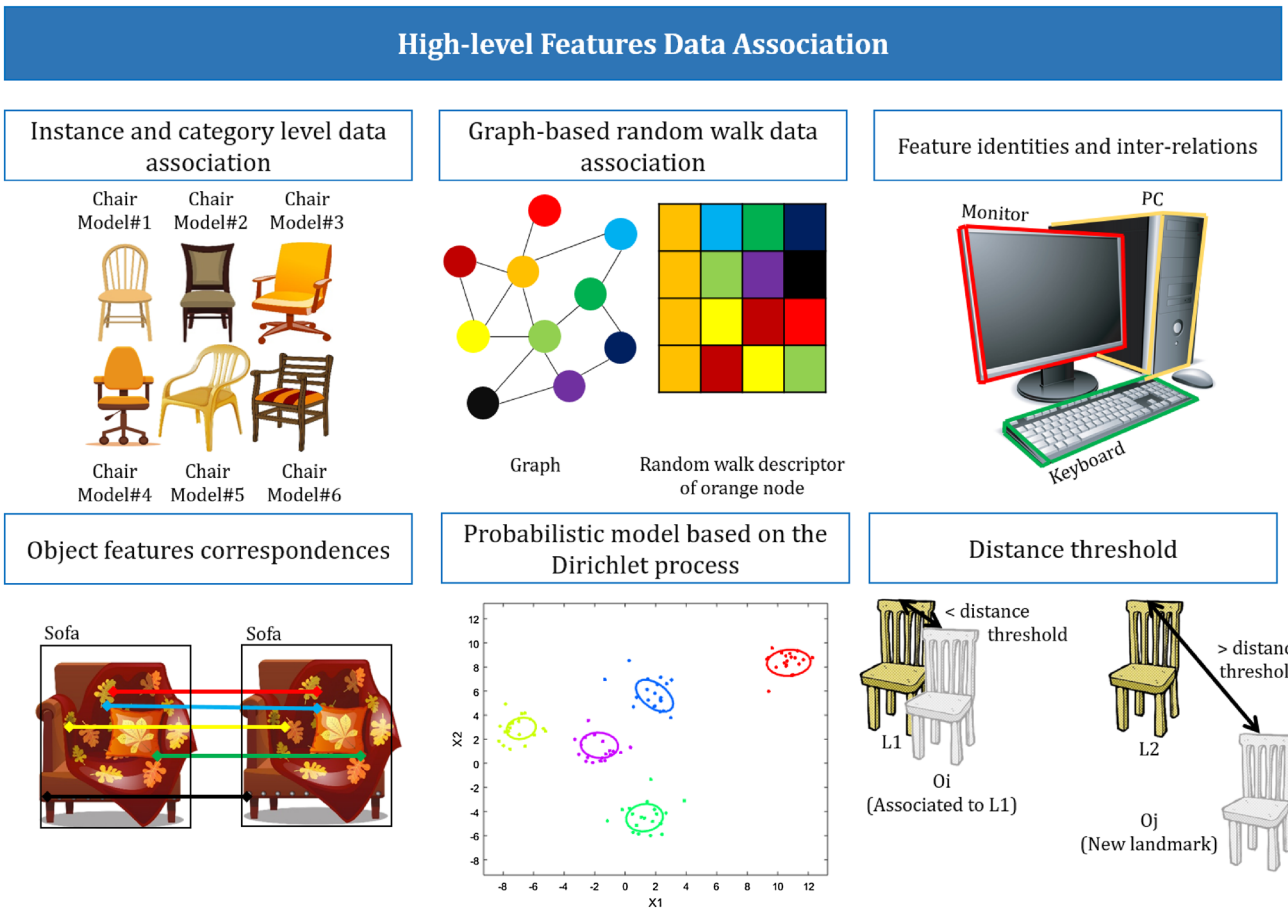


Fig. 11 High-level features data association techniques

discussed techniques that aid the efficiency of high-level feature-based visual SLAM systems is provided in Fig. 12.

3.3.3 Handling dynamics in the scene

The majority of SLAM systems are developed under the unrealistic assumption that the environment is static. Only a few systems were proposed in the literature where scene dynamics are accounted for. Most of these system detect the non-stationary parts of the observed scene, eliminates them, then perform SLAM based on the remaining static environment. An example of such approach can be found in [5] where moving objects were tracked and stationary ones were used to generate a static map of the environment under investigation. Observations were done using a laser scanner and data association was carried out using a multi-level RANSAC approach.

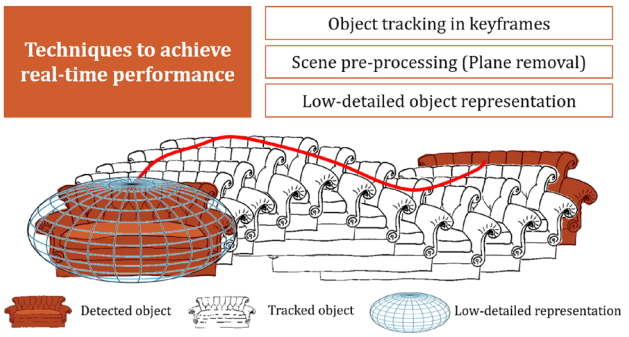


Fig. 12 Techniques to achieve real-time performance by high-level feature-based visual SLAM approaches

Differently, the work presented in [137] uses cuboids as objects' representation, where an object SLAM system is proposed. The system relies on observations from a monocular camera and exploits dynamic objects in the scene to improve localization by adding motion model constraints to the multi-view BA formulation that is used to solve the optimization problem. Objects and feature points belonging to them are tracked in successive frames and motion models are estimated and employed to improve the accuracy of trajectory and map estimation.

Exploiting motion models of dynamic objects rather that ignoring them imposes additional constraints on the systems and hence improves the accuracy of the estimation.

A summary of all the reviewed high-level feature-based approaches in the previous sections is provided in Table 3.

3.4 Hybrid feature-based approaches

In the previous sections, SLAM systems that employ features of a single type were discussed and analyzed. Features at each level enhance the outcome of SLAM in

Table 3 Implementation choices adopted by high-level feature-based SLAM approaches

References	SLAM component			
	Initialization	Data association	Loop closure	Estimation
[88]	Open loop predictions	A probabilistic model based on the Dirichlet process	Data association approach	Factor graph
[23]	–	Object class/label within a distance threshold	objects detected by multiple robots	Factor Graph
[45]	Prior Map	Matching of Random-Walk Descriptors	–	Graph-based technique then Maximim A Posteriori
[5]	–	Multi-level RANSAC and Mahalanobis distance	ML-RANSAC	EKF
[69]	Relative poses are estimated then refined using coordinate descent	Contextual Relevance based on the face centric geometric descriptors	Sampling-based approach to search for correspondences	Coordinate descent
[95]	–	Object class/label and greedy object tracker	Hungarian Algorithm (Cost is estimated object shape and pose)	Factor Graph
[137]	Object detections used to initialize depth of points	Matching point features belonging to detected objects, 2-D KLT sparse optical flow algorithm for tracking dynamic points, and visual object tracking for dynamic objects	No explicit loop closure module used	BA

– indicates that implementation details about the corresponding element/block are not provided

a distinct way. For instance, localization methods based on observation of low-level features have achieved a high level of maturity in terms of accuracy and efficiency. The maps they produce, however, are of high sparsity without any semantic indications. Taking advantage of middle-level features, such as planes, in the scene makes it possible to attain higher reconstruction density as well as more robustness in texture-less environments. To create meaningful maps that humans can easily perceive, recent SLAM approaches make effective use of the emerging object detection techniques and employ semantically labeled observations throughout the localization and mapping processes. To make the most out of what can be visually observed in a scene and to enhance their overall outcome, SLAM systems have lately started to employ features at two or more levels as discussed in this section. In this section, feature-based visual SLAM approaches that adopt features from multiple levels will be reviewed. The reviewed systems are classified based on the features used to perform SLAM into three categories; low- and middle-level feature based approaches, low- and high-level feature based approaches, and low-, middle-, and high-level feature-based approaches. Table 4 presents a summary of the implementation choices adopted by the reviewed approaches.

3.4.1 Low- and middle- level feature-based approaches

The systems presented in [28, 52, 67, 138] employ low and middle-level features to achieve the objectives of SLAM.

In some environments, such as corridors, plane SLAM becomes unconstrained. Fusing planes and points can greatly enhance the robustness of SLAM in such environments as proposed in [138] where planes, detected in monocular frames using a pop-up 3D model, are used to estimate the camera trajectory and the 3D map of the environment. Across different frames, planes are associated based on a weighted sum of three quantities: the difference between their normals, the distance between them, and the overlap between their projections. For each incoming monocular frame, ORB descriptors are computed and a bag of words approach is used to detect loops. Upon detection of a loop, corresponding plane pairs are determined and the factor graph is modified accordingly.

Geometric primitives and planes were employed differently in [52]. A least-squares optimization using a graph formulation, where planar constraints are involved, is used to solve the SLAM problem. The detected points are constrained to belong to a particular plane, parameterized by its normal and depth with respect to the camera, in the environment. Angles between planes in the environment are also considered as constraints. All constraints are coupled into a cost function and the resulting non-linear least squares problem is solved.

A third variation was proposed in [67] where an RGB-D SLAM approach based on planes and points was proposed. Each incoming image is divided into intervals, then labeled, based on the planes present in it. The orientation of a frame is estimated based on the orientation of the most dominant plane in it while translation between frames is computed based on matched SIFT features and RANSAC. Global alignment and loop closure are carried out based on a fusion of the low- and middle-level features, which aids the robustness of the proposed approach.

A recent RGB-D SLAM was proposed in [144] where points and planes are exploited to estimate the pose of a camera and a map of its surroundings. ORB features are extracted from RGB frames and handled by the RGB-D version of ORB-SLAM2. On the other hand, depth frames are used to extract planes, along with their contour points from the scene. Contour points are employed to construct spatial and geometric constraints between planes in the reconstructed map. A novel data association technique for planes was used, where the angle between two planes was used to judge whether they are perpendicular or parallel, while accounting for measurement noise. Two planes are matched if the distance between the observed plane's point and the plane in the map is below a particular threshold. Imaginary planes that are perpendicular to planes appearing in the scene are also exploited and treated as the other observed features in the pose estimation process. A factor graph is constructed and solved by means of the Levenberg-Marquardt optimizer. The proposed plane data association method is more robust than approaches considering plane normals and/or plane distances because it takes into account the measurement noise, which is inevitable.

Super-pixels are middle-level features seen as planar regions exhibiting similar intensities in input frames. Employing super-pixels comes with the advantage of being able to reconstruct poorly-textured scenes. However, there isn't a robust descriptor of such features, which makes it hard to match them in different images. In [28], a feature-based monocular SLAM approach was proposed, integrating super-pixels with PTAM, where PTAM keyframes are divided into super-pixels of irregular sizes. The map state, that is to be estimated, consists of the pose of all keyframes, the Euclidean coordinates of point features, and the parameters of the planar super-pixels. Two keyframes, the pose of which is already computed using PTAM, are used to initialize a super-pixel. All super-pixels in the keyframes are extracted and matched using a Monte Carlo approach. BA is used to optimize the camera and 3D points' states, which are then used to estimate the parameters of the super-pixels. On every new keyframe, all

SN Applied Sciences (2020) 2:224 | https://doi.org/10.1007/s42452-020-2001-3

Table 4 Implementation choices adopted by hybrid feature-based SLAM approaches

Ref	SLAM Component					Approach		Features Level		
	Initialization	Data Association	Loop Closure	Estimation	Relocation	Filter	Key-frame	Low	Mid	High
[138]	Ground truth pose	Plane normals' difference, distance between planes and projection overlap	BoW	FG	–	×	×	✓	✓	×
[52]	–	Points belonging to planes are known a priori	–	FG	–	×	×	✓	✓	×
[67]	FF	Joint compatibility branch and bound test	FM	FG	–	×	✓	✓	✓	×
[144]	FF	ORB FM and angles between planes	BoW	FG	FM	×	✓	✓	✓	×
[28]	Monte Carlo Approach, SI & RANSAC	Super-pixel contour re-projection	Done by the mapping module	GBA	–	×	✓	✓	✓	×
[48]	Triangulation	FM between input image and object model using k-d tree search	object matching	GBA	object matching	×	✓	✓	×	✓
[37]	–	SIFT FM + object detection	Implicitly using object detection	Semantic-BA	–	×	×	✓	×	✓
[106]	–	Matching detected text in door signs	Matching detected text in door signs	FG	Matching detected text in door signs	×	×	✓	×	✓
[118]	FF	K-d tree search based on Euclidean distance between 3D points of detection and landmarks	BBW	FG & GBA	BBW	×	✓	✓	×	✓
[26]	IDP	FM then RANSAC	Visual recognition	EKF	Real Time SLAM Relocalization [133]	✓	×	✓	×	✓
[105]	Cheap optimization of the camera position	FAST FM	–	GBA	BBW & RANSAC	×	✓	✓	×	✓
[71]	FM	NNS in a Kd tree and Place Recognition	NNS in a Kd tree and Place Recognition	non-linear LSO	Matching DAISY Descriptors	✓	×	✓	×	✓
[10]	–	Expectation maximization for soft data association	Mahalanobi's distance to semantic measurements	FG	–	×	✓	✓	×	✓
[54]	–	ORB FM, semantic labels & NNS, distance & normal difference between planes	BoW	FG	–	×	✓	✓	✓	✓
[137]	Depth of points initialized based on objects and planes' detections	ORB FM, and matching points of objects and planes	FM	BA	–	×	✓	✓	✓	✓
[90]	Homography or essential matrix and Marker-based initialization [92]	FM, markers matching	Marker detection using a BoW approach	BA	Marker detection OR RANSAC PnP approach	×	✓	✓	✓	×

FG factor graph, *BA* bundle adjustment, *FM* feature matching, *BBW* bags of binary words, *BoW* bag of words, *FF* first frame, *IDP* inverse depth parameterization, *LSO* least square optimization, *NNS* nearest neighbor search

– indicates that implementation details about the corresponding element/block are not provided

✓ indicates that the corresponding option is adopted

× indicates that the corresponding option is not adopted

super-pixels are re-projected to search for matches. When the re-projection error drops below a threshold, the match is added to the optimization problem as a constraint.

Another work that exploits the fusion of point features and planar regions, represented as squared fiducial markers in this case, in an environment can be found in [90]. Besides the robustness achieved due to employing point features, utilizing fiducial markers in this system comes with several advantages such as eliminating scale ambiguity, robustness in repetitive environments where distinguishing point features can be challenging, and feature invariance over time.

3.4.2 Low- and high-level feature-based approaches

A multitude of different SLAM approaches were proposed based on the use of a combination of low- and high-level features in [10, 26, 37, 48, 71, 105, 106, 118, 130]. Such approaches demonstrate high-level expressiveness while maintaining robustness.

The system proposed in [48] does tracking, object recognition, and mapping while mainly operating on monocular RGB frames. Frames exhibiting distinctive geometrical and/or semantic information are selected as keyframes. A semantically labelled object is added to the map after being detected in multiple frames that contain at least 5 point correspondences, have a minimum parallax angle of 3°, and must exhibit acceptable geometric conditioning. To distinguish between instances of the same object model in the scene, the pose of the detected instance in the world frame is hypothesized given the map scale, and the overlap with previously detected instances is computed. If no overlap is detected, a new object instance is added to the map. If the scale of the map is not yet known, objects detected sequentially are assumed to belong to the same object instance in the map. Correspondences are established between measurements and object models using a k-d tree search. For more robustness, ORB features in input images are computed and 2D-3D correspondences are established.

Instead of employing low-level features independently, geometric features can be used to detect objects in the scene as proposed in [37] where object detection and SLAM were done jointly for 2D and 3D sensors using a novel BA formulation, referred to as Semantic BA. Upon reception of a new image, features are extracted and matched to those in the objects model database. A validation graph is then created for each set of correspondences to an object. The frames in which the features are matched along with the model from the database are then transformed into a common pose and the cost of the corresponding semantic feature is the re-projection error of the detected features weighted by the confidence of matches. In the 3D case, when an object is detected multiple times, the cost function of the semantic edge includes the re-projection of one detected feature into the other. Frames in which features were matched to a common point in the model are said to have a virtual match represented by an edge in the graph. For consistency purposes, geometric constraints obtained from SLAM are added to the graph. The resulting validation graph is optimized to obtain the minimum re-projection error for all constraints.

In some environments, such as educational entities and hospitals, each room is assigned a unique identifier which can serve as a landmark in a SLAM system as presented in [106]. After eliminating the points that corresponds to walls, a door-sign detector, based on an SVM classifier, is employed. Characters contained in a door sign are recognized using Optical Character Recognition (OCR). Lines extracted from laser data along with measurements from the door-sign detector are then passed to a mapper to map the environment.

Observations of generic objects were used to extend RGB-D ORB-SLAM2 in [118]. Objects are detected, segmented, and associated to landmarks in the map by means of a k-d tree. The pose of the objects is determined using ORB-SLAM. Detected objects are stored with three pieces of information: the RGB point cloud of the object, their pose from ORB-SLAM, and the accumulated detection confidence. The class label is determined based on the entire history of detection of an object. A sparse map of the environment can be built explicitly by projecting the point cloud based on the latest trajectory estimate. Finally, object points are inserted into the SLAM state vector as Euclidean coordinates and hence are tracked and further refined upon reception of new data in the following frames.

EKF-Monocular-SLAM, Structure from Motion (SfM), and Visual Recognition were combined in the system proposed in [26]. Objects are detected by associating SURF points in an images to object models in a database. Such associations are then geometrically verified using RANSAC. Afterwards, the PnP algorithm or DLT algorithm are employed to compute the transformation or Homography matrices for non-planar and planar models, respectively, which are then used to refine the pose of the object. Matched points are fed into the monocular SLAM module which is based on EKF-Monocular-SLAM where the state vector to be estimated consists of the camera motion parameters and the point features along with the geometry of the detected objects.

On a different note, some scenes in the environment under observation may exhibit dynamicity which if not accounted for, hinders the overall performance of SLAM systems. Hence, most SLAM systems assume a scene where objects remain static throughout the localization

and mapping processes. The SLAM system presented in [105] eliminates this assumption by removing dynamic objects from the observed scene before operation. More specifically, every RGB-D frame is processed to mask out regions in which a person was detected using an RGB-D based method [58]. The remaining data image a static environment which can be processed using a standard visual SLAM algorithm. A similar approach can be found in [130] where dynamic objects are segmented out of the scene by means of a computationally efficient step-wise approach to detect the object and extract its contour. The static environment is then mapped based on point features using a novel look-up table approach that targets using a large amount of distinct, evenly-distributed point features from the environment, which enhances the accuracy of mapping and localization.

Along the same lines, an online method for extracting non-static objects from the observed scene, and hence improving the performance of RGB-D SLAM in non-static environments was proposed in [116]. The approach consists of three main stages, starting with image differencing to detect any moving objects in the scene. A particle filter is then employed to track motion patches in consecutive RGB-D frames, which makes it more general than approaches that track particular object models. Finally, maximum-a-posteriori is used to identify the scene's foreground, after segmenting the moving objects by means of vector quantization. To operate reliably, the approach requires the observed scene to consist mainly of static objects and to contain planes.

As the scene to be re-constructed by visual SLAM grows larger, matching features to points becomes more challenging because some places exhibit similar appearances. To circumvent this, the work presented in [71] employs a coarse place recognition module where frames containing common points are grouped together under location classes using an overlapping view clustering algorithm. Matching features is then done based on the Hamming distance between BRIEF descriptors of Harris corners.

Data association and SLAM are tightly coupled problems that were not considered jointly except in a few research work where they were solved as two optimization sub-problems. Data association for each observation-landmark pair is estimated then used to estimate the sensor and landmark poses. Using this approach, the accuracy of sensor and landmark pose estimation is critically degraded by incorrect data association. In addition, measurements that are discarded due to their ambiguity cannot be reconsidered when more refined measurements of the same landmarks are obtained.

These limitations motivated the changes in the SLAM algorithm proposed in [10] where data association, and estimations of sensor and landmarks poses were considered in a single optimization problem. Instead of associating each observation to a single landmark, Expectation Maximization was employed to account for the entire density of the data association while estimating the sensor and landmark poses, which was referred to as soft data association. Estimation is based on inertial measurements, ORB features, and semantic information obtained from an object detector. The depth of an observed landmark is the median of the ORB features detected within the bounding box of that landmark. In case multiple instances of the same object exist in the environment, Mahalanobi's distance is used to decide data association. An extension of this work was presented in [4], where semantic structure was inferred differently. Instead of relying on ORB features, a stacked hourglass convolutional network was used to detect semantic features of the object found within each bounding box. Structure constraints are used to relate each semantic feature to the corresponding landmark and Kabsch Algorithm is then used to estimate the orientation of the object. A very similar approach can be found in [33] with the distinction that it employs non-Gaussian sensor models as opposed to majority of the proposed approaches, where Gaussian model are always assumed.

The system proposed in [143] combines high-level semantically labeled features and low-level CNN features to localize a mobile robot by means of a coarse to fine approach. Observations are matched to visual frames in the map by first comparing the objects appearing in the image. A finer search is then carried out based on CNN features of the image. The estimated poses of the camera as well as the features are finally refined using BA.

3.4.3 Low-, middle-, and high-level feature-based approaches

In [54] and [137], SLAM systems are developed based on features from all three levels; points, planes, and objects.

The system proposed in [54] employs an RGB-D sensor to observe features in the environment. Real-time, efficient performance of this system is achievable because objects are represented by means of quadrics which do not require highly detailed representation. The SLAM problem is formulated as a factor graph where various types of factors are used, including observations of points, objects, and planes as well as point-plane, plane-plane, and object-plane relationships. A variation of ORB-SLAM2 is used to detect points in the environment which are then matched among frames in a coarse-to-fine pyramid. Faster R-CNN is used to detect objects in incoming frames and the corresponding ellipsoids representing the objects are then computed. Across frames, semantic labels are used to associate observations to objects if a single instance of the object appears in the environment. Otherwise, data

association is achieved by means of nearest-neighbor matching. The point cloud representing the scene is segmented to extract planes using an organized point cloud segmentation technique. Planes are associated using thresholds on the distance between them and the difference between their normals. Factors are added between planes and points that belong to them, objects and the corresponding planes they lie on, and between multiple planes assuming a Manhattan World. A bag of words approach is adopted to detect loop closures.

Using points, planes, and objects, observed through a monocular camera, the work presented in [137] achieves improved localization, especially in absence of loop closure, compared to state-of-the-art SLAM systems. This is attributed to the long-range observability of objects and planes which facilitates more associations between old and new measurements. Objects are represented as cuboids, plane edges are detected then back-projected to obtain their parameters, and points are added to further constrain the camera poses. BA formulation is used with four types of constraints, camera-plane, camera-object, object-plane, and point-plane. The maps generated are dense and exhibit a high level of expressive representation.

4 Conclusion

Simultaneous Localization and Mapping is the most predominant research problem in the robotics community where tremendous amounts of effort are put into generating novel approaches that maximize its robustness and reliability. Upon acquisition of the first set of measurements from the environment to be reconstructed, the trajectory of the robot and the map are initialized. Subsequent measurements go through a pipeline of different processes that are implemented differently in each SLAM system but do achieve the same purpose. Such processes include data association, loop closure, re-localization, and trajectory and map estimation.

In this paper, we surveyed most of the state-of-the-art visual SLAM solutions that employ features to localize the robot and map its surroundings. We classified feature-based visual SLAM approaches into categories based on the types of features they rely on; low-level, middle-level, high-level, or hybrid features. The strengths and weaknesses of each category were thoroughly investigated and the challenges that each solution overcomes were highlighted, when applicable. Comparisons between approaches in the same category were provided in tables, comparing the methods that were adopted to implement each component of the SLAM pipeline.

Based on our intensive review, we believe that the following challenges remain unsolved.

1. *Generality* Current SLAM solutions lack the ability to adapt to the environment in which the robot is operating. Because they depend on a certain type of features. Failure to detect such features in the environment leads to catastrophic degradation in the accuracy of the SLAM outcome. This could be due to the intermittent presence of features in the environment or the inability of the employed vision system to detect them. The former happens if the SLAM system depends on a very limited set of features, for instance the set of objects that a neural network can detect, while not utilizing other elements in the image like planes, geometric primitives, or new objects that the network was not trained to detect. The latter might occur in challenging environments or due to abrupt motions. To cope with such challenges, the vision system employed by SLAM should be flexible to accommodate various types of features based on the environment in which the robot is operating, for example during a transition between indoor and outdoor environments.
2. *Robustness* In presence of noise from several sources in the SLAM pipeline, it is sometimes hard for the estimation algorithm to generate optimum estimates of the map and trajectory. Very limited research work has been done to guarantee the optimality of a SLAM estimate or at least verify whether or not the estimate is optimal [17–19, 55–57]. To that end, post-processing SLAM estimates by means of a neural network, for example, might result in significant improvements to the estimated trajectory and reconstructed map, and hence a more robust SLAM system.
3. *Scene Understanding and Expressive Representation* Ever since the deep learning breakthrough in 2012, object detectors have been heavily exploited in SLAM. However, the current object detectors do not exploit any temporal or spatial relationships between the detections [117]. If such constraints are accounted for, an increase in the efficiency and reliability of the detections is expected.

The advances in software and hardware technology that we currently witness should be directed towards developing an environment-aware, error-free, general visual SLAM algorithm that is capable of circumventing all of these challenges.

Acknowledgements This publication is based upon work supported by the Khalifa University of Science and Technology under Award No. RC1-2018-KUCARS.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Alahi A, Ortiz R, Vandergheynst P (2012) Freak: fast retina keypoint. In: 2012 IEEE conference on computer vision and pattern recognition, pp 510–517. https://doi.org/10.1109/CVPR.2012.6247715
2. Angeli A, Doncieux S, Meyer J, Filliat D (2008) Real-time visual loop-closure detection. In: 2008 IEEE international conference on robotics and automation, pp 1842–1847. https://doi.org/10.1109/ROBOT.2008.4543475
3. Annaiyan A, Olivares-Mendez MA, Voos H (2017) Real-time graph-based slam in unknown environments using a small UAV. In: 2017 international conference on unmanned aircraft systems (ICUAS), pp 1118–1123. https://doi.org/10.1109/ICUAS.2017.7991524
4. Atanasov N, Bowman SL, Daniilidis K, Pappas GJ (2018) A unifying view of geometry, semantics, and data association in slam. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization, pp 5204–5208. https://doi.org/10.24963/ijcai.2018/722
5. Bahraini MS, Bozorg M, Rad AB (2018) Slam in dynamic environments via ml-ransac. Mechatronics 49:105–118. https://doi.org/10.1016/j.mechatronics.2017.12.002
6. Bay H, Ess A, Tuytelaars T, Gool LV (2008) Speeded-up robust features (surf). Similarity matching in computer vision and multimedia. Comput Vis Image Understand 110(3):346–359. https://doi.org/10.1016/j.cviu.2007.09.014
7. Beaudet PR (1978) Rotationally invariant image operators. In: Proceedings of the 4th international joint conference on pattern recognition. Kyoto, pp 579–583
8. Besl PJ, McKay ND (1992) A method for registration of 3-d shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239–256. https://doi.org/10.1109/34.121791
9. Bosse M, Zlot R, Flick P (2012) Zebedee: design of a spring-mounted 3-d range sensor with application to mobile mapping. IEEE Trans Robot 28(5):1104–1119. https://doi.org/10.1109/TRO.2012.2200990
10. Bowman SL, Atanasov N, Daniilidis K, Pappas GJ (2017) Probabilistic data association for semantic slam. In: 2017 IEEE international conference on robotics and automation (ICRA), pp 1722–1729 (2017). https://doi.org/10.1109/ICRA.2017.7989203
11. Brandli C, Berner R, Yang M, Liu S, Delbruck T (2014) A 240 $\times$ 180 130 db 3μs latency global shutter spatiotemporal vision sensor. IEEE J Solid-State Circuits 49(10):2333–2341. https://doi.org/10.1109/JSSC.2014.2342715
12. Bukhori I, Ismail ZH (2017) Detection of kidnapped robot problem in monte carlo localization based on the natural displacement of the robot. Int J Adv Robot Syst 14(4):1729881417717,469. https://doi.org/10.1177/1729881417717469
13. Bultmann S, Li K, Hanebeck U (2019) Stereo visual slam based on unscented dual quaternion filtering. In: Proceedings of the 22nd international conference on information fusion (fusion 2019)
14. Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I, Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. IEEE Trans Robot 32(6):1309–1332. https://doi.org/10.1109/TRO.2016.2624754
15. Caldato BAC, Filho RA, Castanho JEC (2017) Orb-odom: stereo and odometer sensor fusion for simultaneous localization and mapping. In: 2017 latin American robotics symposium (LARS) and 2017 Brazilian symposium on robotics (SBR), pp 1–5. https://doi.org/10.1109/SBR-LARS-R.2017.8215301
16. Calonder M, Lepetit V, Ozuysal M, Trzcinski T, Strecha C, Fua P (2012) Brief: computing a local binary descriptor very fast. IEEE Trans Pattern Anal Mach Intell 34(7):1281–1298. https://doi.org/10.1109/TPAMI.2011.222
17. Carlone L (2013) A convergence analysis for pose graph optimization via Gauss–Newton methods. In: 2013 IEEE international conference on robotics and automation, pp 965–972. https://doi.org/10.1109/ICRA.2013.6630690
18. Carlone L, Dellaert F (2015) Duality-based verification techniques for 2d slam. In: 2015 IEEE international conference on robotics and automation (ICRA), pp 4589–4596. https://doi.org/10.1109/ICRA.2015.7139835
19. Carlone L, Rosen DM, Calafiore G, Leonard JJ, Dellaert F (2015) Lagrangian duality in 3d slam: verification techniques and optimal solutions. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 125–132. https://doi.org/10.1109/IROS.2015.7353364
20. Chen Y, Medioni G (1991) Object modeling by registration of multiple range images. In: Proceedings. 1991 IEEE international conference on robotics and automation, vol 3, pp 2724–2729. https://doi.org/10.1109/ROBOT.1991.132043
21. Cheng J, Wang C, Meng MQ (2019) Robust visual localization in dynamic environments based on sparse motion removal. IEEE Trans Autom Sci Eng. https://doi.org/10.1109/TASE.2019.2940543
22. Cho H, Kim EK, Kim S (2018) Indoor slam application using geometric and icp matching methods based on line features. Robot Auton Syst 100:206–224. https://doi.org/10.1016/j.robot.2017.11.011
23. Choudhary S, Carlone L, Nieto C, Rogers J, Liu Z, Christensen HI, Dellaert F (2017) Multi robot object-based slam. In: Kulić D, Nakamura Y, Khatib O, Venture G (eds) 2016 international symposium on experimental robotics. Springer, Cham, pp 729–741
24. Choudhary S, Trevor AJB, Christensen HI, Dellaert F (2014) Slam with object discovery, modeling and mapping. In: 2014 IEEE/RSJ international conference on intelligent robots and systems, pp 1018–1025. https://doi.org/10.1109/IROS.2014.6942683
25. Civera J, Davison AJ, Montiel JMM (2008) Inverse depth parametrization for monocular slam. IEEE Trans Robot 24(5):932–945. https://doi.org/10.1109/TRO.2008.2003276
26. Civera J, Gálvez-López D, Riazuelo L, Tardós JD, Montiel JMM (2011) Towards semantic slam using a monocular camera. In: 2011 IEEE/RSJ international conference on intelligent robots and systems, pp 1277–1284. https://doi.org/10.1109/IROS.2011.6094648
27. Clipp B, Lim J, Frahm JM, Pollefeys M (2010) Parallel, real-time visual slam. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, pp 3961–3968. https://doi.org/10.1109/IROS.2010.5653696
28. Concha A, Civera J (2014) Using superpixels in monocular slam. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 365–372. https://doi.org/10.1109/ICRA.2014.6906883
29. Concha A, Loianno G, Kumar V, Civera J (2016) Visual-inertial direct slam. In: 2016 IEEE international conference on robotics and automation (ICRA), pp 1331–1338. https://doi.org/10.1109/ICRA.2016.7487266
30. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297. https://doi.org/10.1023/A:1022627411411

31. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol 1, pp 886–893. https://doi.org/10.1109/CVPR.2005.177
32. Davison AJ, Reid ID, Molton ND, Stasse O (2007) Monoslam: real-time single camera slam. IEEE Trans Pattern Anal Mach Intell 29(6):1052–1067. https://doi.org/10.1109/TPAMI.2007.1049
33. Doherty K, Fourie D, Leonard J (2019) Multimodal semantic slam with probabilistic data association. In: 2019 international conference on robotics and automation (ICRA), pp 2419–2425. https://doi.org/10.1109/ICRA.2019.8794244
34. Engel J, Schöps T, Cremers D (2014) Lsd-slam: large-scale direct monocular slam. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer vision—ECCV 2014. Springer, Cham, pp 834–849
35. Engelson SP, McDermott DV (1992) Error correction in mobile robot map learning. In: Proceedings 1992 IEEE international conference on robotics and automation, vol 3, pp 2555–2560. https://doi.org/10.1109/ROBOT.1992.220057
36. Eudes A, Lhuillier M (2009) Error propagations for local bundle adjustment. In: 2009 IEEE conference on computer vision and pattern recognition, pp 2411–2418. https://doi.org/10.1109/CVPR.2009.5206824
37. Fioraio N, Stefano LD (2013) Joint detection, tracking and mapping by semantic bundle adjustment. In: 2013 IEEE conference on computer vision and pattern recognition, pp 1538–1545. https://doi.org/10.1109/CVPR.2013.202
38. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395. https://doi.org/10.1145/358669.358692
39. Flint A, Mei C, Reid I, Murray D (2010) Growing semantically meaningful models for visual slam. In: 2010 IEEE computer society conference on computer vision and pattern recognition, pp 467–474. https://doi.org/10.1109/CVPR.2010.5540176
40. Fuentes-Pacheco J, Ruiz-Ascencio J, Rendón-Mancha JM (2015) Visual simultaneous localization and mapping: a survey. Artif Intell Rev 43(1):55–81. https://doi.org/10.1007/s10462-012-9365-8
41. Gallego G, Delbruck T, Orchard G, Bartolozzi C, Taba B, Censi A, Leutenegger S, Davison A, Conradt J, Daniilidis K, Scaramuzza D (2019) Event-based vision: a survey. CoRR arXiv:1904.08405
42. Galvez-López D, Tardos JD (2012) Bags of binary words for fast place recognition in image sequences. IEEE Trans Robot 28(5):1188–1197. https://doi.org/10.1109/TRO.2012.2197158
43. Gao X, Zhang T (2015). In: 2015 34th Chinese control conference (CCC), pp 5851–5856. https://doi.org/10.1109/ChiCC.2015.7260555
44. Garcia-Fidalgo E, Ortiz A (2015) Vision-based topological mapping and localization methods: a survey. Robot Auton Syst 64:1–20. https://doi.org/10.1016/j.robot.2014.11.009
45. Gawel A, Don CD, Siegwart R, Nieto J, Cadena C (2018) X-view: graph-based semantic multi-view localization. IEEE Robot Autom Lett 3(3):1687–1694. https://doi.org/10.1109/LRA.2018.2801879
46. Gee AP, Chekhlov D, Calway A, Mayol-Cuevas W (2008) Discovering higher level structure in visual slam. IEEE Trans Robot 24(5):980–990. https://doi.org/10.1109/TRO.2008.2004641
47. Gomez-Ojeda R, Moreno FA, Scaramuzza D, Jiménez JG (2017) PL-SLAM: a stereo SLAM system through the combination of points and line segments. CoRR **abs/1705.09479**. arXiv:1705.09479
48. Gálvez-López D, Salas M, Tardós JD, Montiel J (2016) Real-time monocular object slam. Robot Auton Syst 75:435–449. https://doi.org/10.1016/j.robot.2015.08.009
49. Harris C, Stephens M (1988) A combined corner and edge detector. In: In Proceedings of fourth Alvey vision conference, pp 147–151
50. Hartley R, Zisserman A (2003) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, New York
51. He X, Zemel RS, Carreira-Perpinan MA (2004) Multiscale conditional random fields for image labeling. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004, vol 2, pp II–695–II–702. https://doi.org/10.1109/CVPR.2004.1315232
52. Henein M, Abello M, Ila V, Mahony R (2017) Exploring the effect of meta-structural information on the global consistency of slam. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1616–1623. https://doi.org/10.1109/IROS.2017.8205970
53. Ho KL, Newman P (2006) Loop closure detection in slam by combining visual and spatial appearance. Robot Auton Syst 54(9):740–749. https://doi.org/10.1016/j.robot.2006.04.016
54. Hosseinzadeh M, Latif Y, Pham T, Sünderhauf N, Reid ID (2018) Towards semantic SLAM: points, planes and objects. CoRR arXiv:1804.09111
55. Huang S, Dissanayake G (2016) A critique of current developments in simultaneous localization and mapping. Int J Adv Robot Syst 13(5):1729881416669,482. https://doi.org/10.1177/1729881416669482
56. Huang S, Wang H, Frese U, Dissanayake G (2012) On the number of local minima to the point feature based slam problem. In: 2012 IEEE international conference on robotics and automation, pp 2074–2079. https://doi.org/10.1109/ICRA.2012.6224876
57. Huang S, Yingwu Lai, Frese U, Dissanayake G (2010) How far is slam from a linear least squares problem? In: 2010 IEEE/RSJ international conference on intelligent robots and systems, pp 3011–3016. https://doi.org/10.1109/IROS.2010.5652603
58. Jafari OH, Mitzel D, Leibe B (2014) Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 5636–5643. https://doi.org/10.1109/ICRA.2014.6907688
59. Jiang G, Yin L, Jin S, Tian C, Ma X, Ou Y (2019) A simultaneous localization and mapping (slam) framework for 2.5d map building based on low-cost lidar and vision fusion. Appl Sci. https://doi.org/10.3390/app9102105
60. Kaess M, Johannsson H, Roberts R, Ila V, Leonard JJ, Dellaert F (2012) iSAM2: incremental smoothing and mapping using the Bayes tree. Int J Robot Res 31(2):216–235. https://doi.org/10.1177/0278364911430419
61. Kaess M, Ranganathan A, Dellaert F (2008) iSAM: incremental smoothing and mapping. IEEE Trans Robot 24(6):1365–1378. https://doi.org/10.1109/TRO.2008.2006706
62. Kasyanov A, Engelmann F, Stückler J, Leibe B (2017) Keyframe-based visual-inertial online slam with relocalization. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 6662–6669. https://doi.org/10.1109/IROS.2017.8206581
63. Kejriwal N, Kumar S, Shibata T (2016) High performance loop closure detection using bag of word pairs. Robot Auton Syst 77:55–65. https://doi.org/10.1016/j.robot.2015.12.003
64. Kim H, Handa A, Benosman R, Ieng SH, Davison A (2014) Simultaneous mosaicing and tracking with an event camera. In: Proceedings of the British machine vision conference. BMVA Press. https://doi.org/10.5244/C.28.26
65. Klein G, Murray D (2007) Parallel tracking and mapping for small ar workspaces. In: 2007 6th IEEE and ACM international symposium on mixed and augmented reality, pp 225–234. https://doi.org/10.1109/ISMAR.2007.4538852

66. Klein G, Murray D (2008) Improving the agility of keyframe-based slam. In: Forsyth D, Torr P, Zisserman A (eds) Computer vision—ECCV 2008. Springer, Berlin, pp 802–815
67. Le PH, Košecka J (2017) Dense piecewise planar rgb-d slam for indoor environments. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4944–4949. https://doi.org/10.1109/IROS.2017.8206375
68. Leutenegger S, Chli M, Siegwart RY (2011) Brisk: Binary robust invariant scalable keypoints. In: Proceedings of the 2011 international conference on computer vision, ICCV '11, pp 2548–2555. IEEE Computer Society, Washington. https://doi.org/10.1109/ICCV.2011.6126542
69. Li J, Meger D, Dudek G (2017) Context-coherent scenes of objects for camera pose estimation. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 655–660. https://doi.org/10.1109/IROS.2017.8202221
70. Lim H, Lim J, Kim HJ (2014) Real-time 6-dof monocular visual slam in a large-scale environment. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 1532–1539. https://doi.org/10.1109/ICRA.2014.6907055
71. Lim H, Sinha SN, Cohen MF, Uyttendaele M (2012) Real-time image-based 6-dof localization in large-scale environments. In: 2012 IEEE conference on computer vision and pattern recognition, pp 1043–1050. https://doi.org/10.1109/CVPR.2012.6247782
72. Lindeberg T (1998) Feature detection with automatic scale selection. Int J Comput Vis 30(2):79–116. https://doi.org/10.1023/A:1008045108935
73. Liu J, Liu D, Cheng J, Tang Y (2014) Conditional simultaneous localization and mapping: a robust visual slam system. Neurocomputing 145:269–284. https://doi.org/10.1016/j.neucom.2014.05.034
74. Liu W, Anguelov D, Erhan D, Szegedy C, Reed SE, Fu C, Berg AC (2015) SSD: single shot multibox detector. CoRR arXiv:1512.02325
75. Liu Y, Zhang H (2012) Indexing visual features: real-time loop closure detection using a tree structure. In: 2012 IEEE international conference on robotics and automation, pp 3613–3618. https://doi.org/10.1109/ICRA.2012.6224741
76. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision, vol 2, pp 1150–1157. https://doi.org/10.1109/ICCV.1999.790410
77. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
78. Lowe T, Kim S, Cox M (2018) Complementary perception for handheld slam. IEEE Robot Autom Lett 3(2):1104–1111. https://doi.org/10.1109/LRA.2018.2795651
79. Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, Corke P, Milford MJ (2016) Visual place recognition: a survey. IEEE Trans Robot 32(1):1–19. https://doi.org/10.1109/TRO.2015.2496823
80. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th international joint conference on artificial intelligence—vol 2, IJCAI'81, pp 674–679. Morgan Kaufmann Publishers Inc., San Francisco. http://dl.acm.org/citation.cfm?id=1623264.1623280
81. Mair E, Hager GD, Burschka D, Suppa M, Hirzinger G (2010) Adaptive and generic corner detection based on the accelerated segment test. In: Daniilidis K, Maragos P, Paragios N (eds) Computer vision—ECCV 2010. Springer, Berlin, pp 183–196
82. Maity S, Saha A, Bhowmick B (2017) Edge slam: edge points based monocular visual slam. In: 2017 IEEE international conference on computer vision workshops (ICCVW), pp 2408–2417. https://doi.org/10.1109/ICCVW.2017.284
83. Matas J, Chum O, Urban M, Pajdla T (2004) Robust wide-baseline stereo from maximally stable extremal regions. Image Vis Comput 22(10):761–767. https://doi.org/10.1016/j.imavis.2004.02.006
84. Mazuran M, Tipaldi GD, Spinello L, Burgard W, Stachniss C (2014) A statistical measure for map consistency in slam. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 3650–3655. https://doi.org/10.1109/ICRA.2014.6907387
85. Milford MJ, Schill F, Corke P, Mahony R, Wyeth G (2011) Aerial slam with a single camera using visual expectation. In: 2011 IEEE international conference on robotics and automation, pp 2506–2512. https://doi.org/10.1109/ICRA.2011.5980329
86. Milford MJ, Wyeth GF, Prasser D (2004) Ratslam: a hippocampal model for simultaneous localization and mapping. In: Robotics and automation, 2004. Proceedings. 2004 IEEE international conference on ICRA '04, vol 1, pp 403–408. https://doi.org/10.1109/ROBOT.2004.1307183
87. Mouragnon E, Lhuillier M, Dhome M, Dekeyser F, Sayd P (2006) Real time localization and 3d reconstruction. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol 1, pp 363–370. https://doi.org/10.1109/CVPR.2006.236
88. Mu B, Liu SY, Paull L, Leonard J, How JP (2016) Slam with objects using a nonparametric pose graph. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4602–4609. https://doi.org/10.1109/IROS.2016.7759677
89. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: In VISAPP international conference on computer vision theory and applications, pp 331–340
90. Muñoz-Salinas R, Medina Carnicer R (2019) Ucoslam: simultaneous localization and mapping by fusion of keypoints and squared planar markers. CoRR arXiv:1902.03729
91. Mur-Artal R, Tardós JD (2017) Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Trans Robot 33(5):1255–1262. https://doi.org/10.1109/TRO.2017.2705103
92. Muñoz-Salinas R, Marín-Jimenez MJ, Medina-Carnicer R (2019) Spm-slam: simultaneous localization and mapping with squared planar markers. Pattern Recognit 86:156–171. https://doi.org/10.1016/j.patcog.2018.09.003
93. Nicholson L, Milford M, Sünderhauf N (2018) Quadricslam: constrained dual quadrics from object detections as landmarks in semantic SLAM. CoRR arXiv:1804.04011
94. Nitsche MA, Castro GI, Pire T, Fischer T, Cristóforis PD (2017) Constrained-covisibility marginalization for efficient on-board stereo slam. In: 2017 European conference on mobile robots (ECMR), pp 1–6. https://doi.org/10.1109/ECMR.2017.8098655
95. Parkhiya P, Khawad R, Murthy JK, Bhowmick B, Krishna KM (2018) Constructing category-specific models for monocular object-slam. CoRR arXiv:1802.09292
96. Piasco N, Sidibé D, Demonceaux C, Gouet-Brunet V (2018) A survey on visual-based localization: on the benefit of heterogeneous data. Pattern Recogniti 74:90–109. https://doi.org/10.1016/j.patcog.2017.09.013
97. Pire T, Fischer T, Civera J, Cristóforis PD, Berlles JJ (2015) Stereo parallel tracking and mapping for robot localization. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1373–1378. https://doi.org/10.1109/IROS.2015.7353546
98. Posch C, Matolin D, Wohlgenannt R (2011) A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. IEEE J Solid-State Circuits 46(1):259–275. https://doi.org/10.1109/JSSC.2010.2085952

99. Pumarola A, Vakhitov A, Agudo A, Sanfeliu A, Moreno-Noguer F (2017) Pl-slam: real-time monocular visual slam with points and lines. In: 2017 IEEE international conference on robotics and automation (ICRA), pp 4503–4508. https://doi.org/10.1109/ICRA.2017.7989522
100. Qayyum U, Ahsan Q, Mahmood Z (2017) Imu aided rgb-d slam. In: 2017 14th international Bhurban conference on applied sciences and technology (IBCAST), pp 337–341. https://doi.org/10.1109/IBCAST.2017.7868075
101. Qiu K, Liu T, Shen S (2017) Model-based global localization for aerial robots using edge alignment. IEEE Robot Autom Lett 2(3):1256–1263. https://doi.org/10.1109/LRA.2017.2660063
102. Quan M, Piao S, Tan M, Huang S (2019) Accurate monocular visual-inertial slam using a map-assisted ekf approach. IEEE Access 7:34289–34300. https://doi.org/10.1109/ACCESS.2019.2904512
103. Redmon J, Farhadi A (2016) YOLO9000: better, faster, stronger. CoRR arXiv:1612.08242
104. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. CoRR arXiv:1804.02767
105. Riazuelo L, Montano L, Montiel JMM (2017) Semantic visual slam in populated environments. In: 2017 European conference on mobile robots (ECMR), pp 1–7. https://doi.org/10.1109/ECMR.2017.8098697
106. Rogers JG, Trevor AJB, Nieto-Granda C, Christensen HI (2011) Simultaneous localization and mapping with learned object recognition and semantic data association. In: 2011 IEEE/RSJ international conference on intelligent robots and systems, pp 1264–1270. https://doi.org/10.1109/IROS.2011.6095152
107. Rublee E, Rabaud V, Konolige K, Bradski G (2011) Orb: An efficient alternative to sift or surf. In: 2011 international conference on computer vision, pp 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544
108. Sabatini R, Ramasamy S, Gardi A, Rodriguez Salazar L (2013) Low-cost sensors data fusion for small size unmanned aerial vehicles navigation and guidance. Int J Unmanned Syst Eng 1:16–47. https://doi.org/10.14323/ijuseng.2013.11
109. Saputra MRU, Markham A, Trigoni N (2018) Visual slam and structure from motion in dynamic environments: a survey. ACM Comput Surv 51(2):37:1–37:36. https://doi.org/10.1145/3177853
110. Segal A, Hähnel D, Thrun S (2009) Generalized-icp. In: Trinkle J, Matsuoka Y, Castellanos JA (eds) Robotics: science and systems. The MIT Press, Cambridge
111. Shi J, Tomasi C (1994) Good features to track. In: 1994 Proceedings of IEEE conference on computer vision and pattern recognition, pp 593–600. https://doi.org/10.1109/CVPR.1994.323794
112. Shum HY, Szeliski R (2001) Construction of panoramic image mosaics with global and local alignment. Springer, New York, pp 227–268
113. Souto LAV, Nascimento TP (2016) Object subtraction planar rgb-d slam. In: 2016 XIII Latin American robotics symposium and iv brazilian robotics symposium (LARS/SBR), pp 19–24. https://doi.org/10.1109/LARS-SBR.2016.11
114. Stewénius H, Engels C, Nistér D (2006) Recent developments on direct relative orientation. ISPRS J Photogramm Remote Sens 60(4):284–294. https://doi.org/10.1016/j.isprsjprs.2006.03.005
115. Sualeh M, Kim GW (2019) Simultaneous localization and mapping in the epoch of semantics: a survey. Int J Control Autom Syst 17(3):729–742. https://doi.org/10.1007/s12555-018-0130-x
116. Sun Y, Liu M, Meng MQH (2017) Improving rgb-d slam in dynamic environments: a motion removal approach. Robot Auton Syst 89:110–122. https://doi.org/10.1016/j.robot.2016.11.012
117. Sünderhauf N, Brock O, Scheirer W, Hadsell R, Fox D, Leitner J, Upcroft B, Abbeel P, Burgard W, Milford M, Corke P (2018) The limits and potentials of deep learning for robotics. Int J Robot Res 37(4–5):405–420. https://doi.org/10.1177/0278364918770733
118. Sünderhauf N, Pham TT, Latif Y, Milford M, Reid I (2017) Meaningful maps with object-oriented semantic mapping. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5079–5085. https://doi.org/10.1109/IROS.2017.8206392
119. Sünderhauf N, Protzel P (2012) Towards a robust back-end for pose graph slam. In: 2012 IEEE international conference on robotics and automation, pp 1254–1261. https://doi.org/10.1109/ICRA.2012.6224709
120. Taketomi T, Uchiyama H, Ikeda S (2017) Visual slam algorithms: a survey from 2010 to 2016. IPSJ Trans Comput Vis Appl 9(1):16. https://doi.org/10.1186/s41074-017-0027-2
121. Tang J, Ericson L, Folkesson J, Jensfelt P (2019) Gcnv2: efficient correspondence prediction for real-time SLAM. CoRR arXiv:1902.11046
122. Thrun S, Burgard W, Fox D (2005) Probabilistic robotics (intelligent robotics and autonomous agents). The MIT Press, Cambridge
123. Torr P, Zisserman A (2000) Mlesac: a new robust estimator with application to estimating image geometry. Comput Vis Image Understand 78(1):138–156. https://doi.org/10.1006/cviu.1999.0832
124. Trevor AJB, Rogers JG, Christensen HI (2014) Omnimapper: a modular multimodal mapping framework. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 1983–1990. https://doi.org/10.1109/ICRA.2014.6907122
125. Triggs B, McLauchlan PF, Hartley RI, Fitzgibbon AW (2000) Bundle adjustment—a modern synthesis. In: Triggs B, Zisserman A, Szeliski R (eds) Vision algorithms: theory and practice. Springer, Berlin, pp 298–372
126. Unicomb J, Dantanarayana L, Arukgoda J, Ranasinghe R, Dissanayake G, Furukawa T (2017) Distance function based 6dof localization for unmanned aerial vehicles in gps denied environments. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 5292–5297. https://doi.org/10.1109/IROS.2017.8206421
127. Urzua S, Munguía R, Grau A (2017) Vision-based slam system for mavs in gps-denied environments. Int J Micro Air Veh 9(4):283–296. https://doi.org/10.1177/1756829317705325
128. Wang R, Wan W, Wang Y, Di K (2019) A new rgb-d slam method with moving object detection for dynamic indoor scenes. Remote Sens 11:1143
129. Wang X, Vozar S, Olson E (2017) Flag: feature-based localization between air and ground. In: 2017 IEEE international conference on robotics and automation (ICRA), pp 3178–3184. https://doi.org/10.1109/ICRA.2017.7989360
130. Wang Z, Zhang Q, Li J, Zhang S, Liu J (2019) A computationally efficient semantic slam solution for dynamic scenes. Remote Sens 11(11):1363. https://doi.org/10.3390/rs11111363
131. Weikersdorfer D, Adrian DB, Cremers D, Conradt J (2014) Event-based 3d slam with a depth-augmented dynamic vision sensor. In: 2014 IEEE international conference on robotics and automation (ICRA), pp 359–364. https://doi.org/10.1109/ICRA.2014.6906882
132. Weikersdorfer D, Hoffmann R, Conradt J (2013) Simultaneous localization and mapping for event-based vision systems. In: Chen M, Leibe B, Neumann B (eds) Computer vision systems. Springer, Berlin, pp 133–142
133. Williams B, Klein G, Reid I (2007) Real-time slam relocalisation. In: 2007 IEEE 11th international conference on computer vision, pp 1–8. https://doi.org/10.1109/ICCV.2007.4409115

134. Williams R, Konev B, Coenen F (2015) Scalable distributed collaborative tracking and mapping with micro aerial vehicles. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 3092–3097. https://doi.org/10.1109/IROS.2015.7353804
135. Xia Y, Li J, Qi L, Yu H, Dong J (2017) An evaluation of deep learning in loop closure detection for visual slam. In: 2017 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), pp 85–91. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.18
136. Yang S, Maturana D, Scherer S (2016) Real-time 3d scene layout from a single image using convolutional neural networks. In: 2016 IEEE international conference on robotics and automation (ICRA), pp 2183–2189. https://doi.org/10.1109/ICRA.2016.7487368
137. Yang S, Scherer S (2019) Monocular object and plane slam in structured environments. IEEE Robot Autom Lett 4(4):3145–3152. https://doi.org/10.1109/LRA.2019.2924848
138. Yang S, Song Y, Kaess M, Scherer S (2016) Pop-up slam: semantic monocular plane slam for low-texture environments. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1222–1229. https://doi.org/10.1109/IROS.2016.7759204
139. Younes G, Asmar D, Shammas E, Zelek J (2017) Keyframe-based monocular slam: design, survey, and future directions. Robot Auton Syst 98:67–88. https://doi.org/10.1016/j.robot.2017.09.010
140. Younes G, Asmar DC, Shammas EA (2016) A survey on non-filter-based monocular visual SLAM systems. CoRR arXiv:1607.00470
141. Yousif K, Bab-Hadiashar A, Hoseinnezhad R (2015) An overview to visual odometry and visual slam: applications to mobile robotics. Intel Ind Syst 1(4):289–311. https://doi.org/10.1007/s40903-015-0032-7
142. Zhang AS, Liu BS, Zhang CJ, Wang DZ, Wang EX (2017) Fast initialization for feature-based monocular slam. In: 2017 IEEE international conference on image processing (ICIP), pp 2119–2123. https://doi.org/10.1109/ICIP.2017.8296656
143. Zhang W, Liu G, Tian G (2019) A coarse to fine indoor visual localization method using environmental semantic information. IEEE Access 7:21963–21970. https://doi.org/10.1109/ACCESS.2019.2899049
144. Zhang X, Wang W, Qi X, Liao Z, Wei R (2019) Point-plane slam using supposed planes for indoor environments. Sensors 19:3795
145. Zhao L, Huang S, Sun Y, Yan L, Dissanayake G (2015) Parallaxba: bundle adjustment using parallax angle feature parametrization. Int J Robot Res 34(4–5):493–516. https://doi.org/10.1177/0278364914551583