Advanced Robotics, Vol. 21, No. 3–4, pp. 233–265 (2007)

Also available online - www.brill.nl/ar

Review

Recent advances in simultaneous localization and map-building using computer vision

ZHENHE CHEN *, JAGATH SAMARABANDU and RANGA RODRIGO

University of Western Ontario, Department of Electrical and Computer Engineering,
1151 Richmond Street North, London, Ontario N6A 5B9, Canada

Received 10 February 2006; accepted 25 April 2006

Abstract—Simultaneous localization and map-building (SLAM) continues to draw considerable attention in the robotics community due to the advantages it can offer in building autonomous robots. It examines the ability of an autonomous robot starting in an unknown environment to incrementally build an environment map and simultaneously localize itself within this map. Recent advances in computer vision have contributed a whole class of solutions for the challenge of SLAM. This paper surveys contemporary progress in SLAM algorithms, especially those using computer vision as main sensing means, i.e., visual SLAM. We categorize and introduce these visual SLAM techniques with four main frameworks: Kalman filter (KF)-based, particle filter (PF)-based, expectation-maximization (EM)-based and set membership-based schemes. Important topics of SLAM involving different frameworks are also presented. This article complements other surveys in this field by being current as well as reviewing a large body of research in the area of vision-based SLAM, which has not been covered. It clearly identifies the inherent relationship between the state estimation *via* the KF *versus* PF and EM techniques, all of which are derivations of Bayes rule. In addition to the probabilistic methods in other surveys, non-probabilistic approaches are also covered.

Keywords: Robot localization; map-building; computer vision; probabilistic frameworks; set membership.

1. INTRODUCTION

An autonomous mobile robot is an intelligent agent which explores an unknown environment with minimal human intervention. Building a relative map which describes the spatial model of the environment is essential for exploration by such a robot. Possessing a map with sufficient location information of landmarks and obstacles can make it possible for the robot to estimate its pose, to plan its path and

*To whom correspondence should be addressed. E-mail: zchen56@uwo.ca

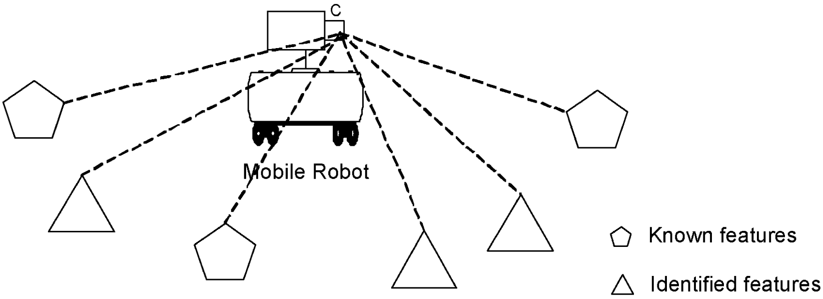

Figure 1. A depiction of feature-based SLAM.

to avoid collisions. Conversely, if pose of the robot is given all along its trajectory, the map can be easily acquired through its perception of the outside world.

The procedure of simultaneous localization and map-building (SLAM) can be described as follows (Fig. 1). A robot starts its navigation in an unknown environment from an unknown location. The robot navigates using its dead-reckoning sensor (e.g., odometer). As other onboard sensors perceive features (landmarks) from the environment, the SLAM estimator performs a series of processing: recognize the feature, determine whether it is a new one, calculate its spatial position and add it to current feature map. Concurrently, the estimator localizes the robot according to identified and known features. In this way, the robot can incrementally build the environment feature map and localize itself.

The ability to estimate both the map and the robot location is usually due to formulating the statistical correlations that exist between the estimates of the position of the robot and landmarks, and between those of the landmarks themselves. Generally in SLAM, the robot is subject to errors in measurement and motion control. A strategy for increasing map-building accuracy is to re-observe identified or ‘old’ landmarks. Meanwhile, the robot pose uncertainty is also constrained by this re-observation.

Robotic map-building can be traced back to 25 years ago. At the beginning, they were generally grouped into metric [1] and topological approaches [2, 3]. Since the 1990s, probabilistic approaches have become dominant in map-building. An important milestone of map-building is a series of papers presented by Smith *et al.* [4, 5]. They presented a powerful probabilistic framework for SLAM. Encouraged by their success, a large number of probability-based solutions to SLAM were published [6–20]. Most of them used probabilistic frameworks to process the measurement noise (sensing errors) and turn the measurements into a map. Only a few used non-probabilistic techniques [21, 22].

The physical environments explored in these recent studies are very diverse. For example, Kim *et al.* [7, 12] studied airborne SLAM in unknown terrain. Newman [23] explored the sub-sea domain. Sujan *et al.* [15, 16] investigated planetary environments, such as cliffs and Martian terrain. However, most of the algorithms and applications were for navigation in structured indoor environments

[9, 10, 17, 19, 21, 22, 24]. While many of these algorithms were used on a single robotic platform, some of them employed groups of robots to corporate during the navigation procedure [11, 14, 21, 22, 25].

As indicated above, sensors are the main avenue for robots to perceive their environment. Commonly used sensors can be categorized into laser-based, sonar-based, inertial-based and vision-based systems. Laser ranging systems are active sensors that are accurate, but their point-to-point measurement characteristic limits the development in semantic object recognition and tracking. Sonar-based systems are fast and cheap, but usually are not very accurate. Some early and successful applications of using sonar for SLAM have been reported in Refs [26, 27]. Sonar provides measurements and recognition capacities similar to vision. However, compared to a large volumes of vision research in SLAM, the use of sonar is limited. In the case of sole dependence on an inertial sensor, such as an odometer, a small error can have large effects on later position estimates [28]. Other short-range sensors, such as infrared and tactile sensors, are not suitable for global measurements, where relatively long-range sensors are required.

One of the mainstream perception techniques is the use of a vision-based sensor. It is desirable for its long range, high resolution and its passive property (i.e., it does not emit energy, which makes it possible to incorporate other heat-sensitive sensors, such as infrared). State-of-the-art research in computer vision has produced several advances that can be exploited in SLAM. Examples include environment map-building in scene modeling, as well as camera motion analysis and description in computer vision. During the last decade, vision-based SLAM solutions have been able to achieve robust solutions mainly due to advances in hardware, mathematics of computer vision and feature abstraction techniques. On the hardware side, both the camera and computer industries have made significant progress so that full-resolution images can be processed at real-time frame rates. In mathematics, the geometry of computer vision has been understood thoroughly and explained systematically only during the past decade [29]. Finally, recent advances in feature extraction enable the usage of high-level vision-based landmarks (complex and natural structures such as doors and road signs) in contrast to early attempts using low-level features (e.g., vertical edges, line segments, etc.) and artificial beacons. As stated by Thrun, SLAM approaches are using greedy algorithms that attempt maximal information gain [28]. Some recently published algorithms can show this type of tendency [7, 8, 10–14, 17, 19, 30, 31]. More literature of computer vision relating to SLAM will be presented in Section 3.

Wolter *et al.* emphasized two important points for SLAM, "any approach to master the SLAM problem can be decomposed into two aspects: handling of map features (extraction from sensor data and matching against the (partially) existing map) and handling of uncertainty" [32]. Accordingly, computer vision techniques can handle feature tracking and recognition well, and they provide robust three-dimensional (3-D) reconstruction which is essential for constructing a feature map. Meanwhile, probabilistic estimation theories offer many paradigms to optimize uncertainty and error. Therefore, the recent trend of state-of-the-art SLAM algorithms is to use computer vision as a perception mechanism within a probabilistic framework.

We would like to add this survey to two additionally related surveys. The first, by De Souza *et al.*, focused mainly on map-based localization with a limited coverage of map-building pertaining to publications up to late 1990s [33]. The second, by Thrun, provided a comprehensive review of robotic map-building with a focus on probabilistic map-building algorithms without specifying any type of sensors [28]. It also included an adequate review of 53 map-building applications. With these two surveys as the backdrop, we would like to focus on recent research activity in SLAM using computer vision and refer to additional 49 applications and 28 computer vision algorithms not included in previous surveys. Our survey clearly identifies the inherent relationship between the state estimation *via* Kalman filtering (KF) *versus* particle filtering (PF) and expectation maximization (EM) technique, all of which are derivations of Bayes rule. In addition, three probabilistic methods (i.e., KF based, PF based and EM based) and non-probabilistic approaches (i.e., set membership (SM) based) are covered in our survey.

As a complete literature review in the field of visual SLAM, some key topics that involve different frameworks will also be introduced in this survey. The first and most fundamental one to all SLAM solutions is the data association problem, which arises when landmarks cannot be uniquely identified, and due to this the number of possible hypotheses may grow exponentially [28, 34]. Second, the loop-closing problem requires successful identification of revisited landmarks to build a consistent map, which is a direct application of data association [35]. The third is the bearing-only SLAM. It arises from the limitations of using computer vision in SLAM where it is not possible to calculate a meaningful range from a single measurement. In contrast, cameras can calculate the angle to landmark sightings and apply triangulation to determine an appropriate initial location estimate [30, 36, 37]. The last one is the kidnapped robot problem, which requests a robot to recover from its localization failure [38].

We begin with basic concepts of estimation theory and computer vision. The main frameworks are reviewed in Section 4. We then discuss recent computer vision-based algorithms within the context of these frameworks. Additionally, we analyze their strengths and limitations. Four important topics in SLAM will be discussed after the frameworks. Finally, comparisons of the four classes of solutions are presented.

2. BASIC CONCEPTS IN ESTIMATION THEORIES

2.1. Bayesian recursive estimation

All probabilistic SLAM algorithms are derived from the recursive Bayes rule [39]:

$$p(\mathbf{x}_k|\mathbf{z}^k)p(\mathbf{z}^k) = p(\mathbf{z}^k|\mathbf{x}_k)p(\mathbf{x}_k), \tag{1}$$

where $\mathbf{x}_k$ is the state consisting of the robot pose and of environment features at time k. In visual SLAM, measurements (e.g., snapshot images) of data are obtained over time, $\mathbf{z}^k = \{\mathbf{z}_i, i = 1, \ldots, k\}$ is a set of measurements from time 1 to k, where $\mathbf{z}_k$ is a measurement by robot sensor at time k, which is used to estimate the inaccessible state $\mathbf{x}_k$:

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k), \tag{2}$$

where $\mathbf{h}_k$ is a possibly nonlinear function and $\mathbf{n}_k$ is an independent and identical distributed (i.i.d.) measurement noise sequence.

The process evolution of the state between time $k-1$ and k can be defined by a possibly nonlinear function $\mathbf{f}_k$, such that:

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}), \tag{3}$$

where process noise $\mathbf{w}_{k-1}$ is also i.i.d.

Now let us define the SLAM problem from a Bayesian perspective: it is to recursively calculate some degree of belief in the inaccessible state $\mathbf{x}_k$ given the measurements $\mathbf{z}^k$. Thus, constructing a probability function (PDF) $p(\mathbf{x}_k|\mathbf{z}^k)$, also called a posterior, is necessary. Generally, the PDF can be obtained in a prediction–update recursion.

Consider that a posterior PDF $p(\mathbf{x}_{k-1}|\mathbf{z}^{k-1})$ is given, then the prior of the state at time k can be computed via the Chapman–Kolmogorov equation:

$$p(\mathbf{x}_k|\mathbf{z}^{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}^{k-1})\,\mathrm{d}\mathbf{x}_{k-1}, \tag{4}$$

where the PDF $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is defined by (3). This procedure is the called prediction stage.

In the update stage, a new measurement $\mathbf{z}_k$ is employed to update the prior $p(\mathbf{x}_k|\mathbf{z}^{k-1})$ to determine the posterior $p(\mathbf{x}_k|\mathbf{z}^k)$ *via* the conditional Bayes rule by rewriting (1):

$$p(\mathbf{x}_k|\mathbf{z}^k) = p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{z}^{k-1}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}^{k-1})p(\mathbf{x}_k|\mathbf{z}^{k-1})}{p(\mathbf{z}_k|\mathbf{z}^{k-1})}, \tag{5}$$

where there are three points needed to be further clarified regarding to (5):

- The PDF $p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{z}^{k-1}) = p(\mathbf{x}_k|\mathbf{z}_k)$ is obtained by the fact of a Markov process of order 1.
- A mild assumption is proposed when we were interested in predicting the measurement $\mathbf{z}_k$ given by a known state $\mathbf{x}_k$ and no past measurement provided additional information, a conditional independence could be expressed as follows:

$$p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{z}^{k-1}) = p(\mathbf{z}_k|\mathbf{x}_k).$$

- The factor $p(\mathbf{z}_k|\mathbf{z}^{k-1})$ in the denominator is the same for any value $\mathbf{x}_k$ in the posterior $p(\mathbf{x}_k|\mathbf{z}_k)$. Thus, it is often written as a positive normalizer in Bayes rule and denoted η, such that:

$$\eta = p(\mathbf{z}_k|\mathbf{z}^{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}^{k-1})\,\mathrm{d}\mathbf{x}_k, \tag{6}$$

where the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ is defined by the measurement model in (2) with given statistics of $\mathbf{n}_k$ and prediction $p(\mathbf{x}_k|\mathbf{z}^{k-1})$ is determined in (4).

Therefore, (5) is re-formulated as follows:

$$p(\mathbf{x}_k|\mathbf{z}_k) = \eta p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}^{k-1}). \tag{7}$$

The objective posterior $p(\mathbf{x}_k|\mathbf{z}_k)$ can be solved by multiplying the probability of the measurement $\mathbf{z}_k$ based on the hypothesis of state $\mathbf{x}_k$ and the prior $p(\mathbf{x}_k|\mathbf{z}^{k-1})$. In other words, the recursive Bayesian estimator allows new information to be added simply by multiplying a prior by a current (k-th) likelihood.

Thus, (4) and (7) establish the basis for the optimal Bayesian solution for SLAM. However, such a solution is a conceptual idea that cannot be practically implemented in the real-world. Optimal solutions, such as KF and PF, employing the PDFs in two stages, will be introduced in Section 4.

2.2. Minimum mean-squared error (MMSE) estimation and the posterior PDF

Before moving to the next section, a MMSE approach to determine the state $\mathbf{x}_k$ is shown, which will be used to deduce the relationship between the PDF $p(\mathbf{x}_k|\mathbf{z}^k)$ and optimal filters. First, a cost function for determining an estimate of the state $\hat{\mathbf{x}}_k$ is set as:

$$\hat{\mathbf{x}}_k^{\mathrm{MMSE}} = \arg\min_{\hat{\mathbf{x}}_k} E\{(\hat{\mathbf{x}}_k - \mathbf{x}_k)^{\mathrm{T}}(\hat{\mathbf{x}}_k - \mathbf{x}_k)|\mathbf{z}^k\},$$

where $E\{\cdot\}$ is an expectation operator and T indicates a transpose of a matrix or vector. The motivation for this operation is to find an estimate of the state $\hat{\mathbf{x}}_k$ that minimizes the expected value of the sum of the squared errors between the true $\mathbf{x}_k$ and estimate $\hat{\mathbf{x}}_k$, given all the measurements. It is a classic criterion for optimal estimation in adaptive signal processing [40]. Now, the cost function is specified as follows:

$$J(\hat{\mathbf{x}}_k, \mathbf{x}_k) = \int_{-\infty}^{\infty} (\hat{\mathbf{x}}_k - \mathbf{x}_k)^{\mathrm{T}}(\hat{\mathbf{x}}_k - \mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}^k)\,\mathrm{d}\mathbf{x}_k.$$

Differentiating this cost function, setting to zero and grouping terms, we have:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \int_{-\infty}^{\infty} \mathbf{x}_k p(\mathbf{x}_k|\mathbf{z}^k)\,\mathrm{d}\mathbf{x}_k, \\ \hat{\mathbf{x}}_k^{\mathrm{MMSE}} &= E\{\mathbf{x}_k|\mathbf{z}^k\}. \end{aligned} \tag{8}$$

This result tells us that the MMSE estimate of a random variable given a whole set of measurements is the mean of that variable conditioned on the measurements. In the following sections, these concepts and terms will be employed to specify robotic SLAM algorithms.

3. COMPUTER VISION LITERATURE RELATING TO SLAM

In this section we provide details on how vision is used for robot navigation. First, we distinguish between explicit and implicit use of vision for navigation. Then, we discuss how successful navigation systems tend to use vision in combination with other sensors. Also, we address the problems of feature tracking and 3-D reconstruction, and give important references. Feature tracking and reconstruction are the two important steps that feed the measurements to the SLAM framework.

Vision as a sensor is prominent due it being the 'richest' sense of humans. The ultimate goal is to make the machine see as humans do. There are two major classes of methods that use vision for navigation. The first class uses vision to estimate the 3-D structure (feature locations) and the pose of the robot. Since these methods explicitly calculate the above parameters based on the image information, they are explicit vision methods. The second class uses vision in a rather implicit manner. They do not require an intermediate 3-D reconstruction of the world and/or robot pose estimation to navigate. Instead, changes observed in consecutive images are used in a feedback mechanism to drive the robot. Therefore, these methods are inherently simple, although they may not have the accuracy of the first class of methods.

Systems which use stereo camera pairs and monocular cameras with structure from motion recovery fall into the first category. The main division within this category is due to whether the cameras are calibrated or uncalibrated. Beardsley *et al.* [41] used controlled rotations of an unclaibrated stereo rig to recover the 3-D structure up to an affine transform. They used corner-matching techniques for feature correspondence. Davison's work presented in his thesis [42] dealt with SLAM for a robot with a stereo active head, operating in an unknown environment and using point features in the world as visual landmarks. More SLAM-related details on these are given in Section 4.

Some of the methods falling into the second class of methods are biologically inspired [43]. One common example is 'bee navigation'-type methods based on optical flow. Santos-Victor *et al.* [44] used a two-camera system with opposite optical axes which mimics the centering reflex of a bee. If the camera orientation is tilted toward the robot heading direction, instead of fixing the cameras with opposite optical axes, the motion parameters can give additional structure cues. Giachetti *et al.* [45] described a procedure for obtaining reliable and dense optical flow from image sequences taken by a camera mounted on a car moving in usual outdoor scenarios. They performed dense optical flow estimates using correlation techniques and then estimated the ego-motion. Lerner *et al.* [46] used the optical flow derived from two consecutive camera frames in combination with a digital terrain map to estimate the position, orientation and ego-motion parameters.

3.1. Vision with other sensors

Vehicular intelligent transportation systems (autonomous road navigation systems) tend to use a combination of global positioning systems (GPS) and vision as primary sensors, in addition to many other onboard sensors such as LADARs and inertial sensors. The main use of vision in these systems is not for core navigation. However, they extensively use vision for various purposes worth mentioning in this survey. One of the early examples is the 'NAVLAB' [47] vision system from Carnegie Mellon University. In their system, vision was used for two purposes—road following based on color information and obstacle avoidance in combination with the information from a laser range finder. The main purpose of using vision in 'VIRTUOUS' of Sotelo *et al.* [48] was to to correctly track the lane of any kind of unstructured road (roads without lane markers painted on them), while correctly detecting other vehicles. Successful systems use inertial sensors in combination with the vision. For example 'Stanley' from the Stanford AI group, which won the DARPA challenge in 2005, incorporates GPS measurements, a 6-d.o.f. inertial measurement unit and the wheel speed for pose estimation. While this vehicle is in motion, the environment is perceived through four laser range finders, a radar system, a stereo camera pair and a monocular vision system. Lobo and Dias [49] used the vertical reference provided by an inertial sensor to estimate the pose of the stereo rig, which makes the reconstruction problem simple. They also describe in detail the methods of using inertial sensor data in a vision system.

3.2. Feature tracking and 3-D reconstruction

There are two major problems to be solved in order to exploit the richness of vision for robot localization and map-building: the feature recognition and tracking problem, and the 3-D reconstruction problem. Feature tracking is the problem of estimating the locations of features in an image sequence. Although there is a mature body of literature addressing feature recognition [50–53] and tracking [54–56], the problem itself is still considered unsolved. For example, a combination of Harris corner detector and RANSAC is a reasonably good tracker. For a recent comparison of corner detectors please refer to Kenney *et al.* [57]. Scale-invariant feature transforms (SIFT) proposed by Lowe [58] are a highly discriminative class of features. They facilitate matching across image pairs with strong affine warps and wide baseline matching. Once the features are tracked, 3-D reconstruction can be performed. The 3-D reconstruction is the problem of obtaining the 3-D coordinates and the camera (robot) pose using two or more (2-D) images by using the understanding of the geometry of multiple image formation. The typical scenario consists of a camera mounted on a robot platform, making images, obtaining the 3-D reconstruction using these images and guiding the robot.

The 3-D reconstruction problem is solved based on the understanding on multiple view geometry and this subject has been comprehensively addressed in classic books by Ma *et al.* [29], Hartley and Zizerman [59] and Faugeras *et al.* [60]. The

3-D reconstruction can be categorized into two, i.e., calibrated and uncalibrated reconstruction, and reviews can be found in Lu *et al.* [61] and Fusiello [62]. Two views are related by the epipolar geometry and a special matrix called the fundamental matrix (essential matrix when the cameras are calibrated). In addition to the references given above on multiple-view geometry, a comprehensive review on the estimation of epipolar geometry is found in Zhang [63]. One method of obtaining the 3-D structure using multiple views is the factorization method introduced by Tomasi and Kanade [64–67]. The scene structure obtained using the multiple-view geometry techniques relates to the true structure only up to a transform (affine or projective [59]). Upgrading this structure to a similarity transform is called metric reconstruction, Euclidean reconstruction, stratification or normalization [68]. However additional constraints such as landmarks are required to upgrade the structure to the true coordinates. If the sequence of (i) feature tracking, (ii) 3-D reconstruction and (iii) normalization and upgrading to true structure is followed, the robot location and the scene structure is available for navigation, with the only uncertainty due to inherent noise contamination.

3.3. Limitations of using vision

There are many drawbacks in using vision and realizing the naive perception (making the computer see as the humans do) is a daunting task. What this implies is that there are many practical difficulties in adopting vision for navigation. Humans seem to rely on a vast amount of prior experience in making decisions based on what their eyes see. Pattern recognition and machine learning for vision are important in this context. In any case, we make a lot of assumptions about the environment or the vision system itself in using vision for navigation. For example, we may assume that the robot is placed in the indoor environment, and that there are a lot of geometrical structures present and detectors such as the Harris corner detector that are able to find trackable features. Feature detection and tracking themselves are intimidating. When there is a region of no-texture, e.g., white wall or a grassy land seen from afar, trackers simply fail. On the other hand, when there is repeating structure, e.g., bricks on a brick wall, and when no global context is used in tracking, there is no method of finding true matches. Therefore, the requirements of the problem are 2-fold: there should be features and they should be distinct. Even if the tracking problems are solved, there are complications in assimilating structure and pose from these measurements. Tracked features may form degenerative configurations, although rare in practice, so that no true structure recovery is possible. Moreover, although well understood, reconstruction from motion parallax is a mathematically complex and numerically sensitive problem. On top of all these, measurements done via images are inherently noisy. In summary, vision is a difficult problem with many drawbacks. This does not mean that it is not useable for navigation. If used with care, vision can produce amazing results, and readers are strongly encouraged to read Hartley and Zisserman [59] and Ma *et al.* [29].

4. APPROACHES WITHIN SLAM FRAMEWORKS

The recent advances in SLAM by computer vision are implemented within estimation frameworks, most of which are probabilistic. These frameworks can be categorized into four groups: KF, PF, and EM in the probabilistic category and SM in the non-probabilistic category.

4.1. KF-based algorithms

KF has a 40-year history for estimating the state of a linear dynamic system perturbed by Gaussian white noise using measurements that are linear functions of the system state, but corrupted by additive Gaussian white noise [69]. From the mid-1980s, Smith *et al.* introduced KF to the field of robotic map-building [4, 5]. So far, an overwhelming number of solutions to the SLAM problem have been implemented within the framework of KF or one of its variants.

Being well known as a good study paradigm for Bayesian prediction–update recurrence, KF requires three assumptions:

- Equation (3) must be a known linear state transition function $\mathbf{F}_k$ in its arguments with added process Gaussian noise $\mathbf{w}_{k-1}$, such that:

$$\mathbf{x}_k = \mathbf{F}_k\mathbf{x}_{k-1} + \mathbf{w}_{k-1}.$$

- Similarly, (2) must be a known linear measurement function $\mathbf{H}_k$ in its arguments with added measurement Gaussian noise $\mathbf{n}_k$:

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{n}_k.$$

- The initial posterior $p(\mathbf{x}_0|\mathbf{z}^0) = p(\mathbf{x}_0)$ must be Gaussian.

These three assumptions can guarantee the posterior $p(\mathbf{x}_k|\mathbf{z}^k)$ is always a Gaussian [70].

The reason for Gaussianity is due to the fact that the KF algorithm can be viewed as the following relationship:

$$\begin{aligned} p(\mathbf{x}_{k-1}|\mathbf{z}^{k-1}) &= \mathcal{N}(\mathbf{x}_{k-1}; \boldsymbol{\mu}_{k-1|k-1}, \boldsymbol{\Sigma}_{k-1|k-1}) \\ p(\mathbf{x}_k|\mathbf{z}^{k-1}) &= \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_{k|k-1}, \boldsymbol{\Sigma}_{k|k-1}) \\ p(\mathbf{x}_k|\mathbf{z}^k) &= \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_{k|k}, \boldsymbol{\Sigma}_{k|k}), \end{aligned}$$

where $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a Gaussian PDF with argument (the state) $\mathbf{x}$, mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The subscripts $_{k-1|k-1}$, $_{k|k-1}$ and $_{k|k}$ represent last time step *a posteriori*, current *a priori* and current *a posteriori*, respectively [40]. Thus, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are predicted and updated in two stages:

- Prediction:

$$\boldsymbol{\mu}_{k|k-1} = \mathbf{F}_k\boldsymbol{\mu}_{k-1|k-1} \tag{9}$$

$$\boldsymbol{\Sigma}_{k|k-1} = \mathbf{F}_k\boldsymbol{\Sigma}_{k-1|k-1}\mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_{k-1}, \tag{10}$$

where $\mathbf{Q}_{k-1}$ is the covariance of $\mathbf{w}_{k-1}$ that is assumed to be zero mean and independent to $\mathbf{n}_k$.

- Update:

$$\boldsymbol{\mu}_{k|k} = \boldsymbol{\mu}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\boldsymbol{\mu}_{k|k-1}) \tag{11}$$
$$\boldsymbol{\Sigma}_{k|k} = \boldsymbol{\Sigma}_{k|k-1} - \mathbf{K}_k\mathbf{H}_k\boldsymbol{\Sigma}_{k|k-1}, \tag{12}$$

where

$$\mathbf{S}_k = \mathbf{H}_k\boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \tag{13}$$
$$\mathbf{K}_k = \boldsymbol{\Sigma}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{S}_k^{-1}, \tag{14}$$

are the covariances of the innovation term $\boldsymbol{\nu}_k = \mathbf{z}_k - \mathbf{H}_k\boldsymbol{\mu}_{k|k-1}$ and Kalman gain, respectively. $\mathbf{R}_k$ is the covariance of $\mathbf{n}_k$ with zero mean.

However, there are two important clarifications for the above formulation:

- The reasons for using the mean $\boldsymbol{\mu}_k$ of state $\mathbf{x}_k$ as the substitute of the estimate state $\hat{\mathbf{x}}_k$ can be found in (8).
- Some applications denote the process of determining $\boldsymbol{\nu}_k$ as the 'observation' stage.

A complete mathematical derivation of KF can be found in Ref. [40]. Interested researchers are referred for further details.

4.1.1. Variants of KF in SLAM. There are two main variants of KF in state-of-the-art SLAM research, i.e., the extended Kalman filtering (EKF) and the sparse extended information filtering (SEIF), which will be introduced in the following two separate subsections.

4.1.2. EKF. Commonly, non-linearities exist in the SLAM problem. For example, the control command in SLAM usually contains trigonometric functions. Then the robot pose does not depend linearly on the previous pose. To accommodate this, the process model in KF has to be modified as follows:

$$\boldsymbol{\mu}_{k|k-1} = \mathbf{f}_k(\boldsymbol{\mu}_{k-1|k-1}),$$

where $\mathbf{f}_k(\cdot)$ is a non-linear state transition function of the previous state $\mu_{k-1|k-1}$. Another non-linearity lies in sensor measurement in SLAM, i.e., the measurement model is rewritten as follows:

$$\mathbf{z}_{k|k-1} = \mathbf{h}_k(\boldsymbol{\mu}_{k|k-1}),$$

where $\mathbf{h}_k(\cdot)$ is a non-linear measurement function of the state prediction and relies on the sensor properties. To meet the assumptions of KF, $\mathbf{f}_k(\cdot)$ and $\mathbf{h}_k(\cdot)$ are

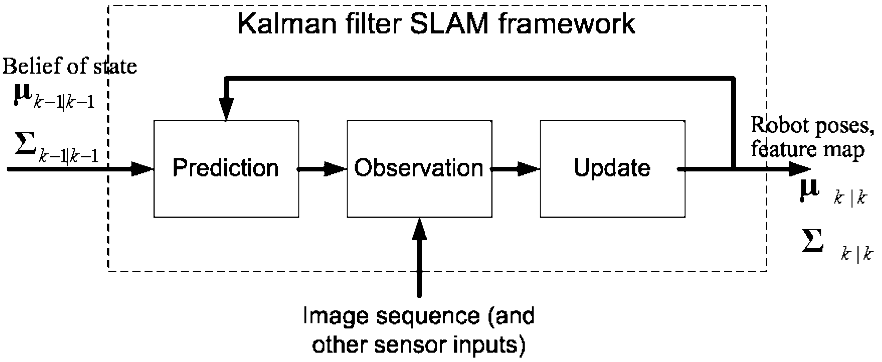
Kalman filter SLAM framework
Belief of state
$\boldsymbol{\mu}_{k-1|k-1}$
$\boldsymbol{\Sigma}_{k-1|k-1}$
Prediction
Observation
Update
Robot poses, feature map
$\boldsymbol{\mu}_{k|k}$
$\boldsymbol{\Sigma}_{k|k}$
Image sequence (and other sensor inputs)

Figure 2. Flow chart of KF-based SLAM.

linearized by first-order Taylor series expansions [71] as follows:

$$\begin{aligned}\hat{\mathbf{F}}_k &= \left.\frac{\mathrm{d}\mathbf{f}_k(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\mathbf{x}=\boldsymbol{\mu}_{k-1|k-1}}\\ \hat{\mathbf{H}}_k &= \left.\frac{\mathrm{d}\mathbf{h}_k(\mathbf{x})}{\mathrm{d}\mathbf{x}}\right|_{\mathbf{x}=\boldsymbol{\mu}_{k|k-1}},\end{aligned} \tag{15}$$

where the local $\hat{\mathbf{F}}_k$ and $\hat{\mathbf{H}}_k$ are to replace their counterparts in (9)–(14). The generalizing filter is known as an EKF. An illustration of an EKF-based SLAM framework is shown in Fig. 2.

In the field of SLAM, researchers recently employed EKF combining novel models of computer vision and environments. For example, Davison *et al.* presented a SLAM approach using active vision [8, 72]. They mounted only one active stereo camera to solve the SLAM problem in structured indoor environments. Like most of the SLAM algorithms, inputs were from an odometer and feature point depth from computer vision. Typically, the outputs were robot pose and a 3-D feature map. We will skip such descriptions in the remaining part of this paper unless they are different. The strengths of this algorithm lie in its active vision sensors, i.e., active vision can act and interact with the environments [73]. In other words, the cameras can observe the feature points from many directions so that the vehicle does not need to adjust its pose much when observing feature points. In this algorithm, feature-tracking and matching approaches can handle ‘natural’ features (e.g., physical objects) in a lab environment. The limitations include the inability to work well in a large open space, since it is only suitable for topological feature-rich information environments. Real experiments were reported [8, 72] where a robot transverses a go–return trip with a 24 m trajectory. The localization estimate can be accurate up to the centimeter level.

Castellanos *et al.* used the symmetries and perturbations model (SPmodel) to represent the environments. They fused a 2-D laser rangefinder and a charge coupled device (CCD) trinocular stereo system that gained redundancy to achieve sensing reliability [9, 74]. Self-calibrations of the laser and stereo camera were given, and their mutual calibration was also provided. The laser data readings were segmented to supporting lines and their length estimated (counted from endpoints); these readings were upgraded to high-level features (corners and semi-planes) by fusing laser rangefinder with vision data. Vision detected the edges. Processed sensor results were fused utilizing multi-sensory calibration and the SPmodel, and then corresponding features were paired. The benefits of this fusion are to maximize both sensors advantages, e.g., the laser does well on range measurement and vision detects accurate edges. Of course, the weakness of computational complexity comes from combining uncertainties in the measurement of two types of sensors and from redundancy removal. Experimental results show that this multisensor-based approach was 96.2% compatible to ground truth.

Tomatis *et al.* used a similar sensing system, laser scanner and CCD camera for their global topological and local metric algorithm to SLAM [10, 75]. Extracted features were divided into two groups for topological and metric environment models, which were corners and openings for the former, and lines for the latter. EKF was employed in the metric model. Such a hybrid strategy provides benefits in computing efficiency since global topological representation demands less computation. However, switching from topological to metric is brittle and needs to be improved.

Se *et al.* proposed a vision-based SLAM algorithm by tracking the SIFT visual landmarks in an indoor unmodified environment [13, 76, 77]. The authors introduced a novel algorithm to select, match and track visual landmarks so that features were invariant to image transform, e.g., translation, scaling and rotation, and partially invariant to illumination, affine or 3-D projection. Processed features with 3-D spatial information were stored in the SIFT database and concurrent pose estimation was performed by ego-motion estimator. EKF was applied to reduce uncertainties of the stored features when compared to current features in the case of revisiting. Some practical issues in SLAM were also considered, e.g., feature viewpoint and occlusion were determined by maintaining a view direction for each landmark.

Kim *et al.* utilized a stereo camera and incorporated other sensors on a fixed wing fight platform to explore unknown terrain environments [7, 78, 79]. These papers addressed the first trials for the airborne SLAM problem using an actual flight vehicle and real data. Such a dynamic airborne platform introduced more motion and measurement formulating difficulties, e.g., fast flight speeds (40 m/s), 6 d.o.f. in the 3-D environment and excessive vibration that affected inertial drift rate. The algorithm formulated a dynamically nonlinear vehicle model by a strapdown inertial navigation system (INS) that represented the position, velocity and altitude of the platform [80, 81]. An inertial measurement unit (IMU) provided state prediction. Meanwhile, the system was equipped with a vision sensor to determine actual range, bearing and elevation to update parameters with the EKF framework. From 50 landmarks in the test-field, the vision sensor detected and registered 19 of those landmarks. All experimental results were verified by INS and GPS. An average landmark accuracy of 7.6 m with 5 m initial uncertainty was reported in their studies. Hygounenc *et al.* were also interested on airborne vehicle SLAM. They built a wide base stereo vision bench on their airship so as to build an elevation map [12]. Their algorithm was similar to that of Se *et al.* [13]. The stereo vision computed 3-D Cartesian coordinates of the perceived pixels. Interest point selection and matching method could identify visually 'natural' landmarks (e.g., a car) from consecutive aerial images. The products of stereo vision and interest point matching offered the estimation of six displacement parameters between the images. This estimation could reject errors from matching, then compute an accurate estimate of the motion between consecutive images. Motion estimation was then refined by EKF. Experimental results showed that the translation errors were below 0.1 m along a 60 m trajectory, while angular errors were below 0.5°.

A number of researchers deal with the SLAM problem using teams of robots. Cooperative behavior in SLAM is driven by the benefits from multiple robots. For example, multiple robots can localize themselves more efficiently if they exchange position information and sense each other. Additionally, a group of cheap robots can acquire redundant information in the environment being explored, which gives higher fault tolerance than one powerful and expensive robot. Madhavan *et al.* addressed localization and terrain map-building algorithms for a distributed cooperative outdoor multi-robot. These robots localized their poses within the EKF framework, and combine the elevation gradient and vision-based range according to those poses to acquire a local map. These multiple local maps were merged during specific motion segments into a globally consistent metric map [11]. Other notable contributions in the field of team robotic map-building are as follows: Sujan *et al.* proposed schemes for robot teams so that the robots could efficiently map a planetary environment, where sensor uncertainty and occlusions were significant [15, 16]. Fierro *et al.* designed a framework and the software architecture for the deployment of multi-robots in an unstructured and unknown environment [14]. Each of these robots had an omnidirectional camera as the sole sensor. Landmarks and other robots could be identified by a YUV color space-based feature extractor, which provided robustness to variation in illumination. An omnidirectional camera produced a range map and range as well as bearing to the localizer. Robot poses were determined within the EKF framework.

One of the recent advances in visual SLAM within the EKF framework is to use monocular vision. This is due to the fact that at every snapshot moment, stereo vision systems need to process two or more images, select features from these images and match associated features from each other. In contrast, a single camera only needs to process one image at every snapshot and match features every two consecutive images, i.e., the single-camera system performs more efficiently. For instance, Davison presented a single-camera algorithm for SLAM [30, 82]. However, his technique depended on some assumptions, which included a calibrated camera, known starting point and smooth camera motion. This motion was constrained with constant velocity and constant angular velocity motion that could be modeled by probability. Chen and Samarabandu proposed a scheme to visual SLAM, which implemented MVG within the EKF framework [83]. The MVG algorithm provided accurate structure and motion measurements from a monocular camera, whereas traditional vision-based approaches require stereo vision. It showed that the proposed algorithm could avoid the limitations of using MVG alone. Thanks to MVG, the algorithm could be easily applied to single- or multiple-camera sensing systems.

4.1.3. SEIF. Another key variant form of KF is IF, which is implemented by propagating the inverse of the state error covariance matrix Σ_k. Such an inverse is related to Fisher's information matrix and is interpreted as a filter in information-theoretical terms [40].

Two terms denote correspondence to their counterparts in KF: information matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ and the information vector $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$. Thus, the IF algorithm can be summarized as follows:

- Prediction:

$$\boldsymbol{\Omega}_{k|k-1} = (\mathbf{F}_k \boldsymbol{\Omega}^{-1}_{k-1|k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{Q}_{k-1})^{-1}$$
$$\boldsymbol{\xi}_{k|k-1} = \boldsymbol{\Omega}_{k|k-1} (\mathbf{F}_k \boldsymbol{\Omega}^{-1}_{k-1|k-1} \boldsymbol{\xi}_{k-1|k-1}).$$

- Update:

$$\boldsymbol{\xi}_{k|k} = \boldsymbol{\xi}_{k|k-1} + \mathbf{H}_k^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{z}_k$$
$$\boldsymbol{\Omega}_{k|k} = \boldsymbol{\Omega}_{k|k-1} + \mathbf{H}_k^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{H}_k.$$

Similar to the necessity of EKF, if using the same denotations in EKF, the EIF is formulated as follows [84]:

- Prediction:

$$\boldsymbol{\mu}_{k-1|k-1} = \boldsymbol{\Omega}^{-1}_{k-1|k-1} \boldsymbol{\xi}_{k-1|k-1} \tag{16}$$
$$\boldsymbol{\Omega}_{k|k-1} = (\hat{\mathbf{F}}_k \boldsymbol{\Omega}^{-1}_{k-1|k-1} \hat{\mathbf{F}}_k^{\mathrm{T}} + \mathbf{Q}_{k-1})^{-1} \tag{17}$$
$$\boldsymbol{\xi}_{k|k-1} = \boldsymbol{\Omega}_{k|k-1} \boldsymbol{\mu}_{k|k-1} = \boldsymbol{\Omega}_{k|k-1} \mathbf{f}_k(\mu_{k-1|k-1}).$$

- Update:

$$\boldsymbol{\xi}_{k|k} = \boldsymbol{\xi}_{k|k-1} + \hat{\mathbf{H}}_k^{\mathrm{T}} \mathbf{R}_k^{-1} [\mathbf{z}_k - \mathbf{h}_k(\boldsymbol{\mu}_{k|k-1}) - \hat{\mathbf{H}}_k \boldsymbol{\mu}_{k|k-1}] \tag{18}$$
$$\boldsymbol{\Omega}_{k|k} = \boldsymbol{\Omega}_{k|k-1} + \hat{\mathbf{H}}_k^{\mathrm{T}} \mathbf{R}_k^{-1} \hat{\mathbf{H}}_k. \tag{19}$$

There are two main advantages of the IF over the KF. First, if in the update stage, there are N ($N > 1$) sensor data obtained at the k-th moment, and these data can be filtered by simply summing the information matrix and vector. Thus, the

(18) and (19) are modified as:

$$\boldsymbol{\xi}_{k|k} = \boldsymbol{\xi}_{k|k-1} + \sum_{i}^{N} \hat{\mathbf{H}}_{k,i}^{\mathrm{T}} \mathbf{R}_k^{-1} [\mathbf{z}_{k,i} - \mathbf{h}_{k,i}(\boldsymbol{\mu}_{k|k-1}) - \hat{\mathbf{H}}_{k,i} \boldsymbol{\mu}_{k|k-1}] \tag{20}$$

$$\boldsymbol{\Omega}_{k|k} = \boldsymbol{\Omega}_{k|k-1} + \sum_{i}^{N} \hat{\mathbf{H}}_{k,i}^{\mathrm{T}} \mathbf{R}_k^{-1} \hat{\mathbf{H}}_{k,i}, \tag{21}$$

where $\mathbf{h}_{k,i}$ and $\hat{\mathbf{H}}_{k,i}$ are the measurement function to the i-th feature and its Jacobian, respectively. This advantage makes the EIF fit for the multi-robot problem: robots provide decentral data so that the EIF can integrate them to calculate a more accurate estimate [25]. Second, IF is naturally more stable than the KF [84].

However, the main problem hinders the applications of the EIF to SLAM is that there are more matrix inversions in EIF ((16) and (17)) than in EKF. Thus, EIF is generally believed to be computationally expensive. Thrun *et al.* proposed a fully on-line SLAM algorithm by sparsifying the information matrix $\boldsymbol{\Omega}_k$, i.e., SEIF [25, 85]. The key principle making SEIF superior to conventional the EKF is the constant time results existing in the SEIF:

- The measurements can be incorporated into SEIF in constant time. It is a natural property of the EIF, which can be understood by the measurement update in (20) and (21).
- If $\boldsymbol{\Omega}_k$ is sparse and the Jacobian of robot pose change $\hat{\mathbf{F}}_k$ in (15) is zero, the constant time of motion update is guaranteed.
- If $\boldsymbol{\Omega}_k$ is sparse, but $\hat{\mathbf{F}}_k$ in equation (15) is nonzero, an optimization method is applied to approximate matrix inversion so that the mean $\boldsymbol{\mu}_{k-1|k-1}$ is available, which is named the amortized constant-time coordinate descent algorithm.

All these results in update of SEIF are constant time so that the processing time of the SEIF algorithm is independent of the size of the map. It it important to note that a prerequisite of SEIF is the sparseness of the information matrix $\boldsymbol{\Omega}_k$. However, $\boldsymbol{\Omega}_k$ is naturally not sparse. Thus, a sparsification technique is applied to remove the link between a feature and the robot. Such a feature is deactivated, and the links between ‘active’ features and the robot are updated to compensate for the removal. The removal result depends on the magnitude of the link before removal. Thrun *et al.* addressed a constant-time sparsification technique to solve the dependance. The technique approximates the posterior $p(\mathbf{x}_k, \mathbf{Y}|\mathbf{z}^k)$, where $\mathbf{Y}$ is the set of all features including three subsets, i.e., the set of all active features, features being deactivated and deactivated features. Once the posterior is approximated correctly, the information matrix $\boldsymbol{\Omega}_k$ is ensured to be sparse at all times.

Overall, there are four steps executed in sequence in the loop of the SEIF SLAM algorithm: motion update, state estimate update, measurement update and sparsification if motion or measurement update violating the sparseness constraint. Details of the algorithm can be found in Ref. [84].

Both real environment experiments and numerical simulation are performed in Ref. [85]. In the real environment experiments, the vehicle transverses 3.5 km and the average position error is smaller than 0.5 m. Compared to EKF, SEIF is almost twice as fast and uses a quarter less memory than that of EKF. In the simulation part of the experiments, three series of experiments are used to examine data association, error comparisons between EKF and SEIF, and multi-robot SLAM. All series of experiments result in satisfying values. The noteworthy point is that SEIF generates bigger error than EKF, which offsets part of the advantages of speed and computation. However, when the size of the map dimension increases largely, the advantages are salient.

4.2. PF-based methods

PF, also called the sequential Monte-Carlo (SMC) method, is a recursive Bayesian filter that is implemented in Monte Carlo simulations [86]. It executes SMC estimation by a set of random point clusters ('particles') representing the Bayesian posterior. In contrast to parametric filters (e.g., KF), PF represents the distribution by a set of samples drawn from this distribution. There are two significant virtues for PF compared to other Bayesian filters:

- Since this representation is approximate and nonparametric, the distributions it can represent are not limited to Gaussian.
- This sampling-based representation can model nonlinear transformations very well.

The second point is particularly useful when handling the use of highly nonlinear sensor and robot motion models, whilst EKF is derived from the first-order Taylor series expansions [71] and has difficulties in high linearity cases. However, like a double-edge sword, this sampling-based approximation suffers from its growth of computational complexity with the state dimension [86], whilst in SLAM, the state is usually composed of both the robot pose and hundreds of features, which makes it impossible to implement PF in practical real-time applications. Therefore, in the state-of-the-art SLAM research, PF has only been successfully applied to localization, but not to map-building [17, 31, 34, 87]. In this section, we would like to review PF literature with only a brief introduction of the underlying mathematics. Interested readers are referred to the book by Ristic *et al.* [86] for more details.

Denote a set of particles $\mathbf{X}_k = \{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(N)}\}$, where each particle $\mathbf{x}_k^{(i)}$ $(1 \leqslant i \leqslant N)$ is a hypothesis of the state $\mathbf{x}_k$ at time k. N, usually a large number, is the number of particles. Ideally, the hypothesis $\mathbf{x}_k^{(i)}$ shall be proportional to the Bayesian posterior:

$$\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{z}^k).$$

A basic PF algorithm can be summarized in Table 1. Here, the important weight $w_k^{(i)}$ arises by the principle of importance sampling [86], which can be interpreted as follows. In order to compute a PDF f, but only given samples generated from a

Table 1.
Basic PF algorithm

```
(1) Inputs: $\mathbf{X}_{k-1}$ and $\mathbf{z}^k$
(2) $\mathbf{X}_k = \emptyset$
(3) For $i = 1 : N$
    • Sample (prediction) $\mathbf{x}_k^{(i)} \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(i)})$
    • Important weight estimation: $w_k^{(i)} = p(\mathbf{z}_k|\mathbf{x}_k^{(i)})$
    • Update: $\mathbf{X}_k = \mathbf{X}_k \cup \{\mathbf{x}_k^{(i)}, w_k^{(i)}\}$
(4) End For
```

different PDF g, a weighting factor w is constructed that $w(x) = f(x)/g(x)$, which counts for the mismatch between f and g. f is termed the 'target distribution' and g as the 'proposal distribution', such that

$$\begin{aligned} f = p(\mathbf{x}^{k,(i)}|\mathbf{z}^k) &\overset{\text{Bayes}}{=} \eta p(\mathbf{z}_k|\mathbf{x}^{k,(i)}, \mathbf{z}^{k-1}) p(\mathbf{x}^{k,(i)}|\mathbf{z}^{k-1}) \\ &\overset{\text{Markov}}{=} \eta p(\mathbf{z}_k|\mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}) p(\mathbf{x}^{k-1,(i)}|\mathbf{z}^{k-1}) \end{aligned} \tag{22}$$

and:

$$g = p(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}) p(\mathbf{x}^{k-1,(i)}|\mathbf{z}^{k-1}). \tag{23}$$

So that:

$$\begin{aligned} w_k^{(i)} &= \frac{\text{target distribution}}{\text{proposal distribution}} = \frac{f}{g} \\ &= \eta p(\mathbf{z}_k|\mathbf{x}_k^{(i)}), \end{aligned} \tag{24}$$

where the constant η has no impact in important sampling since the posterior of the sampling particles is proportional to $w_k^{(i)}$. Now we can rewrite set $\mathbf{X}_k$ yielding

$$\begin{aligned} \mathbf{X}_k &= \{\mathbf{x}_k^{(i)}, w_k^{(i)}|i = 1, \dots, N\} \\ &= \{\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(N)}, w_k^{(1)}, \dots, w_k^{(N)}\}. \end{aligned} \tag{25}$$

Similar to other Bayesian estimation methods, PF estimates up-to-current-step posterior $p(\mathbf{x}^k|\mathbf{z}^k)$ recursively from up-to-last-step posterior $p(\mathbf{x}^{k-1}|\mathbf{z}^{k-1})$. When such posterior is represented by a set of particles, the PF constructs the particle set $\mathbf{X}_k$ recursively from the last-step particle set $\mathbf{X}_{k-1}$. In additional to the basic algorithm depicted in Table 1, practical PF approaches have to solve degeneracy, choice of importance density and resampling problems. A practical way to compute the importance weight can be found in Ref. [31].

Figure 3 shows a general diagram for PF localization. As argued above, there is not important paper using PF for both localization and map-building. For example, Murphy used a Rao-Blackwellized PF to solve a simple form of the SLAM problem

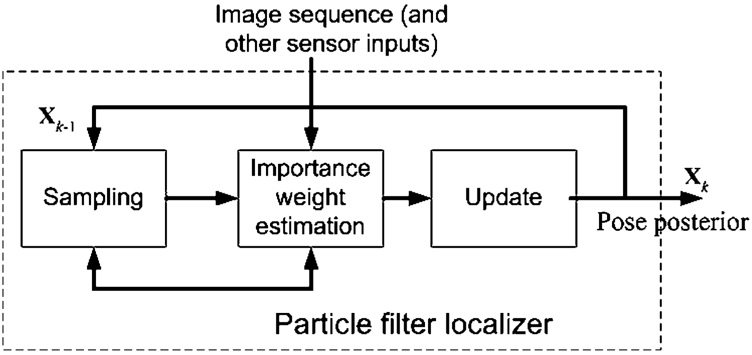
Image sequence (and other sensor inputs)
$\mathbf{x}_{k-1}$
Sampling
Importance weight estimation
Update
$\mathbf{x}_k$
Pose posterior
Particle filter localizer

Figure 3. Flowchart of the PF localizer.

with a solution domain limited to a small 10 × 10 grid world [88]. However, Thrun *et al.* proposed a PF–EKF-based algorithm for FastSLAM solution, where PF was for robot pose and EKF estimated landmark locations based on the estimate poses [31, 34]. They clearly showed that if there were totally K landmarks and M poses ($K \gg M$ and K up to 10^6), their FastSLAM algorithm could reduce running time from $O(K^2)$ in EKF to $O(MK)$ in PF–EKF. Their proposed tree-based data structure could further reduce it to $O(M \log K)$. Additionally, this FastSLAM can solve both online SLAM problems, i.e., determining the PDF $p(\mathbf{x}_k|\mathbf{z}^k)$, and full SLAM problems, i.e., determining the PDF $p(\mathbf{x}^k|\mathbf{z}^k)$ via (22)–(24).

Porta *et al.* presented a vision-based localization approach within the PF framework [17, 89]. In these papers, appearance-based feature representation was introduced, which could simplify computation, and enhance resistance to noise, occlusion and changes in illumination [90]. Such representation could avoid the difficulties of geometric counterparts (e.g., polygonal obstacles with shape and position information, landmarks), such as complexity and proneness to errors. In contrast to other SLAM algorithms, the authors utilized onboard active stereo vision to acquire the disparity map first. After map-building, they presented a novel method to compress such a map to a reduced set of features. Principle component analysis (PCA) was applied for compressing. Additionally, the EM algorithm was used to deal with missing values. Processed results were the preliminarily selected features in disparity map and intensity images. A sensor fusion mechanism was designed for final feature selection. This fusion as well as localization was performed by PF. Another interesting point proposed by this paper lies in an entropy-based active vision strategy, which can solve the 'next-best-view' problem in visual SLAM [91]. Results in real experiments on an actual vehicle were very promising with localization errors in the range of ±25 cm and ±5°.

4.3. EM-based algorithms

Like KF and PF, EM is a general-purpose algorithm for estimation which has received the attention of the SLAM research community. In the light of maximum likelihood (ML) estimation, this technique exploits the incomplete-data problem and offers an optimal solution, which makes it an ideal candidate for map-building. However, this technique needs to process the same data repeatly to obtain the most likely map. In other words, it does not perform efficiently and incrementally. Consequently, it is not a real-time algorithm and is unsuitable for online SLAM [28, 92]. Practical applications employed an incremental ML approach, one part of EM, to construct the map when the robot's path is given by other techniques [93]. In this section, we will provide the basic mathematical background for the EM algorithm in the context of robot navigation and highlight the important contributions in this area.

Suppose the robot path $\mathbf{x}^k = \{\mathbf{x}_1, \ldots, \mathbf{x}_k\}$ and measurement data $\mathbf{z}^k = \{\mathbf{z}_1, \ldots, \mathbf{z}_k\}$ are given, the objective is to evaluate the $(i + 1)$-th map $\mathbf{m}^{(i+1)}$ yielding the EM algorithm, such that

$$\mathbf{m}^{(i+1)} = \arg\max_{\mathbf{m}} E_{\mathbf{x}^k}\{\log p(\mathbf{z}^k, \mathbf{x}^k|\mathbf{m})|\mathbf{m}^{(i)}, \mathbf{z}^k\}. \tag{26}$$

Equation (26) can be interpreted as two iterations of the EM algorithm:

- Expectation step (E-step): $E_{\mathbf{x}^k}\{\log p(\mathbf{z}^k, \mathbf{x}^k|\mathbf{m})|\mathbf{m}^{(i)}, \mathbf{z}^k\}$, where $\mathbf{x}^k$ is the target in this step, $\mathbf{z}^k$ is a complete set and given, and map $\mathbf{m}$ is given and a complete set up to the i-th component. $\log p(\mathbf{z}^k, \mathbf{x}^k|\mathbf{m})$ implies that the log-likelihood is formed for $\mathbf{x}^k$ if $\mathbf{z}^k$ is fully observed and the i-th map $\mathbf{m}^{(i)}$ is given. The purpose of the expectation operation $E\{\cdot\}$ is to calculate the next moment map $\mathbf{m}^{(i+1)}$ given by the log-likelihood of the full path (pose) conditioned on the current moment map $\mathbf{m}^{(i)}$ and observation data $\mathbf{z}^k$.
- Maximization step (M-step): algorithm maximizes the most likely map given pose expectation.

There are two points in the E-step that we need to clarify. First, $\mathbf{x}^k$ (pose) is unknown. Thus, (26) is re-written under the above mild assumptions [28], such that

$$\mathbf{m}^{(i+1)} = \arg\max_{\mathbf{m}} \sum_t \int p(\mathbf{x}_t|\mathbf{m}^{(i)}, \mathbf{z}^k) \log p(\mathbf{z}_t|\mathbf{x}_t, \mathbf{m})\, \mathrm{d}\mathbf{x}_t,$$

where the term $p(\mathbf{x}_t|\mathbf{m}^{(i)}, \mathbf{z}^k)$ is the localization problem that we encountered before. Solving this term becomes the second important point in the E-step. Suppose $t < k$, we need to run Bayes rule twice to compute $p(\mathbf{x}_t|\mathbf{m}^{(i)}, \mathbf{z}^k)$: one is from time 1 to t, $p(\mathbf{x}_t|\mathbf{m}^{(i)}, \mathbf{z}^t)$; another is to find the remaining $p(\mathbf{x}_t|\mathbf{m}^{(i)}, \mathbf{z}^{t+1}, \ldots, \mathbf{z}^k)$. After multiplying two terms and normalizing, the desired posterior $p(\mathbf{x}_t|\mathbf{m}^{(i)}, \mathbf{z}^k)$ is obtained. The E-step calculates expectations for different poses at all points in each moment. This re-localization method makes EM non-incremental. Surprisingly, such re-localization can tackle the correspondence problem (i.e., data association problem), which arises when different features in the environment look alike [34]. This is a key advantage of EM over KF. More details will be given in Section 5.1.

Some recent SLAM algorithms only use the M-step of the EM approach, commonly called incremental ML, to map the environments, while localization is realized by other probabilistic techniques. For instance, Thrun contributed a SLAM algorithm for a team of robots which build a map online and accommodates large odometry errors [93]. The PF-based localizer coped with odometer readings to estimate the poses. Concurrently, the incremental ML utilized a laser rangefinder and panoramic images to build the map. Extensions of map-building in this approach included 3-D restructuring and texture mapping.

Another application of EM in robotic map-building was to describe environments by basic geometric shapes or objects, such as lines, walls, etc. For example, Jogan *et al.* proposed a vision-based robot localization framework in Refs [19, 94]. Two models called learning and localization models were implemented within this framework. In the learning model (similar to map-building), the EM technique was employed for simplifying the obtained panoramic images. Results were used by the localizing mechanism to match stored landmarks and compute the robot poses. Likewise, Thrun *et al.* developed a series of algorithms in this field [93, 95, 96]. However, those developments were beyond of scope of the visual SLAM. Interested researchers are referred those articles for further details.

4.4. SM-based approaches

Apart from probabilistic frameworks that are used to formulate SLAM uncertainty, SM estimation theory also tackles such uncertainty. In contrast to statistical assumption on uncertainty, e.g., correlation modeling in KF, SM imposes an assumption that uncertainty is bounded in norm by some quantity. Estimates of robot and landmark positions are defined by those regions where the robot and landmarks are guaranteed to lie, according to given information. These estimates of regions are termed feasible uncertainty sets [97]. In this section, we briefly review the basic formulation of SM-SLAM and discuss the computer vision usage in the applications.

SM can be applied to either single-robot SLAM or the multi-robot case. Here, we only present single-robot formulations and slight extensions to the multi-robot case. In a 2-D environment, robot motion at the k-th moment is described by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k + \mathbf{G}_k \mathbf{w}_k,$$

where robot pose $\mathbf{x}_k = [x_k y_k \theta_k]^{\mathrm{T}}$ is represented by x–y coordinates and one orientation parameter, $\mathbf{w}_k$ is the error affecting motion, $\mathbf{u}_k$ is the input control vector provided by external driving commands, and $\mathbf{G}_k$ is a matrix to shape this noise. Noise applied to absolute orientation with respect to (w.r.t.) the world coordinate system is scalar and formulated as follows

$$\Theta_k = \theta_k + v_{\theta,k},$$

where $v_{\theta,k}$ is the noise affecting the absolute orientation measurement. Now the state vector $\mathbf{S}_k$, including robot pose as well as n selected landmarks $\mathbf{L}_i = [x_{L_i} y_{L_i}]^{\mathrm{T}}$

$(i = 1, \ldots, n)$, is formulated as follows:

$$\mathbf{S}_k = \left[\mathbf{x}_k^{\mathrm{T}}, L_1^{\mathrm{T}}, \ldots, L_n^{\mathrm{T}}\right]^{\mathrm{T}}.$$

Then, when the landmarks are static, the updated state yields:

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{E}_3\mathbf{u}_k + \mathbf{E}_3\mathbf{G}_k\mathbf{w}_k, \tag{27}$$

where $\mathbf{E}_3 = [\mathbf{I}_3\mathbf{0}]^{\mathrm{T}} \in \mathbb{R}^{3+2n}$ and $\mathbf{I}_3$ is a 3×3 identity matrix. Two sets of measurement equations take on the form:

$$\begin{aligned} \triangle_{i,k} &= d_i(\mathbf{S}_k) + v_{d_i,k} \\ A_{i,k} &= \alpha_i(\mathbf{S}_k) + v_{\alpha_i,k}, \end{aligned} \tag{28}$$

where $\triangle_{i,k}$ and $A_{i,k}$ are actual range and orientation readings provided by the sensors, respectively; relevant range noise is $v_{d_i,k}$, and orientation noise is $v_{\alpha_i,k}$. More specifically, $d_i(\mathbf{S}_k)$ represents distance from the robot pose to the i-th landmark w.r.t. world coordinate system and $\alpha_i(\mathbf{S}_k)$ is the orientation between these two parties w.r.t. the world coordinate system. So far the equations are very similar to the context of KF/EKF. However, in SM, the errors are not formulated by statistics, e.g., a correlation matrix, but by assuming that such errors are unknown but bounded (UBB) [97], such that

$$\begin{aligned} &\|\mathbf{w}_k\|_\infty \leqslant \epsilon_k^{w} \\ &|v_{\theta,k}| \leqslant \epsilon_k^{v_\theta} \\ &|v_{d_i,k}| \leqslant \epsilon_k^{v_d} \\ &|v_{\alpha_i,k}| \leqslant \epsilon_k^{v_\alpha}, \end{aligned} \tag{29}$$

where $\epsilon_k^{w}, \epsilon_k^{v_\theta}, \epsilon_k^{v_d}$ and $\epsilon_k^{v_\alpha}$ are known positive scalars. $\|\cdot\|_\infty$ denotes the l_∞ norm, e.g., $\|v\|_\infty = \max v_i$ is the maximal noise over all v_i. Then, these terms can be embedded into measurement sets $\mathcal{M}_k$ if the sensor readings $\triangle_{i,k}$ and $A_{i,k}$ are given, such that:

$$\mathcal{M}_k = \mathcal{M}_{o,k} \bigcap \left\{ \bigcap_{i=1}^{n} \mathcal{M}_{i,k} \right\}, \tag{30}$$

where at the k-th moment, orientation prediction set $\mathcal{M}_{o,k}$ includes all valid absolute orientation measurements Θ_k whose errors are not greater than $\epsilon_k^{v_\theta}$ and yields

$$\mathcal{M}_{o,k} = \{\mathbf{S}\colon\ |\Theta_k - \theta| \leqslant \epsilon_k^{v_\theta}\},$$

and at the k-th moment, actual range and orientation measurement sets w.r.t. the i-th landmark include all valid measurements whose errors are not greater than $\epsilon_k^{v_d}$ and $\epsilon_k^{v_\alpha}$, respectively, and yield

$$\mathcal{M}_{i,k} = \left\{\mathbf{S}\colon\ |\triangle_{i,k} - d_i(\mathbf{S})| \leqslant \epsilon_k^{v_d}, |A_{i,k} - \alpha_i(\mathbf{S})| \leqslant \epsilon_k^{v_\alpha}\right\}.$$

Let $\Xi_{k|k}$ denote a feasible state set, which is the set of state vector $\mathbf{S}_k$ compatible with robot dynamics given by the process model in (27), measurement equations in (28). Then, we combine all terms and equations for the single-robot SM-SLAM recursive solution, such that:

$$\Xi_{0|0} = \Xi_0 \tag{31}$$

$$\Xi_{k|k-1} = \Xi_{k-1|k-1} + \mathbf{E}_3\mathbf{u}_{k-1} + \mathbf{E}_3\mathbf{G}_{k-1}\epsilon^{w}_{k-1}\mathcal{B}_\infty \tag{32}$$

$$\Xi_{k|k} = \Xi_{k|k-1} \bigcap \mathcal{M}_k, \tag{33}$$

where $\mathcal{B}_\infty$ is the unit ball in $\|\mathbf{w}_k\|_\infty$. Equation (31) is the initial state of the robot containing the initial pose and landmark position. Equation (32) acquires information from an odometer or other inertial sensor describing robot motion. Equation (33) combines all other valid measurements from the measurement sets constrained by the conditions of UBB.

Figure 4 depicts the procedure of SM-SLAM. However, $\mathcal{M}_k$ in (30) cannot be computed since $\mathcal{M}_{o,k}$ and $\{\bigcap_{i=1}^{n} \mathcal{M}_{i,k}\}$ are not compatible: the former is ‘non-convex’ and the latter is non-linear. In Refs [21, 22], Di Marco *et al.* proposed a set of approximations to approach this problem. The basic idea is to choose a class of simple approximation region $\mathcal{R}$ (e.g., box, ellipse, etc.) and select an element $\mathcal{R}_{k|k}$ in the class at each moment k, where $\Xi_{k|k} \in \mathcal{R}_{k|k}$. This selection must meet some requirements so that the algorithm is efficient and suitable for real-time implementation. For example, the size of region $\mathcal{R}$ must be minimized and the computation complexity must be kept small.

SM-SLAM can be extended reasonably well to multi-robot systems. The extensions are from more formulated measurement sets. These include measurements of inter-robots, and measurements between the robot and the landmark; uncertainty affecting the measurements between the robot and the landmark, and affecting the measurements of inter-robots and similar measurement sets for orientation.

In the implementation of SM-SLAM reported by Di Marco *et al.* [21, 22], an omnidirectional stereo system was equipped on the multi-robots. Range data

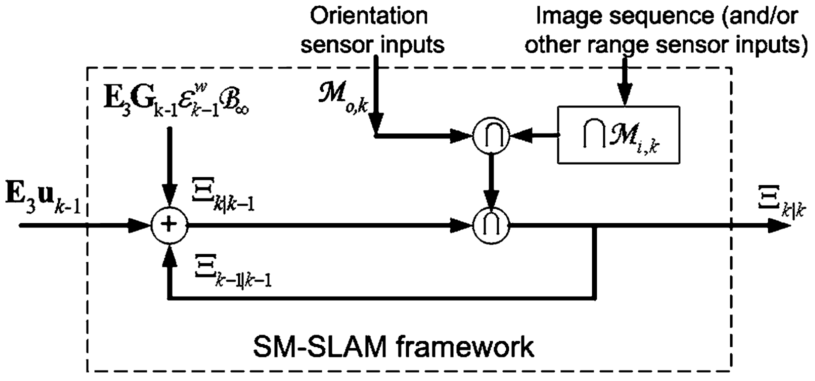
Orientation sensor inputs
Image sequence (and/or other range sensor inputs)
$\mathbf{E}_3\mathbf{G}_{k-1}\mathcal{E}^{w}_{k-1}\mathcal{B}_\infty$
$\mathcal{M}_{o,k}$
$\bigcap$
$\bigcap \mathcal{M}_{i,k}$
$\mathbf{E}_3\mathbf{u}_{k-1}$
+
$\Xi_{k|k-1}$
$\bigcap$
$\Xi_{k|k}$
$\Xi_{k-1|k-1}$
SM-SLAM framework

Figure 4. Flowchart of the SM-SLAM framework.

were acquired by vision geometry calculation. Additionally, the authors quantized errors in (29) with empirical constants. For example, they assigned $\epsilon_k^{v_\alpha} = 3°$, $\epsilon_k^{v_d} = \kappa_d d_i^2(\mathbf{S}_k)$ and $\epsilon_k^{v_\theta} \approx 0$, where $\kappa_d = 0.002$, while landmarks are within 50 m. For each robot, motion model error $\mathbf{w}_k$ was generated as independent uniformly distributed signals, with a mean value proportional to the distance transversed during the last-step move. From such error upper-bounds, one can see the accuracies of this technique. Another point summarized from experimental results is that multi-robot SLAM outputs are superior to single ones. The superiorities include consuming time, average landmark uncertainty and average pose uncertainty. So far, only simple 2-D landmarks have been detected by this SM-SLAM approach. Augmenting 'natural' feature analysis would make it more practical and robust.

5. OTHER ISSUES IN SLAM

The main frameworks (KF, PF, EM and SM) deal with noise in the robot's perception and actuation, which is the biggest challenge in the map-building problem [28]. Additionally, other challenges such as perceptual ambiguity (data association problem), loop-closing problem, bearing-only SLAM problem and kidnapped robot problem are also important topics that are worth considering.

5.1. Data association

Data association in SLAM can be simply presented as a feature correspondence problem, which relates two features observed in different positions. Two common applications of such relating (or association) are to match two successive scenes and to close a loop of a long trajectory when a robot comes to the starting point of the trajectory again. Specifically, if an integer n_t is to label the i-th acquired feature f_t^i at time t and if $n_t = j \leqslant N$ where N is the total number of features in the built map, then the i-th feature corresponds to the j-th feature that was observed before. Otherwise, if $n_t > N$, f_t^i is a previously unseen feature.

To determine the value of n_t, a conventional technique is to apply ML. For example, the probability of a data association value n_t can be evaluated through the derivation of Chapman–Kolmogorov, PF and Bayes approaches [34]. By trying all possible observed features, the maximum of the probability result w.r.t. n_t is compared to an empirical threshold and determine whether f_t^i corresponds to an existing j-th feature or it is a previously unseen feature. There are other techniques to deal with the data association problem in SLAM which can be found in Refs [98, 99].

5.2. Loop-closing

The purpose of studying loop-closing is to build a consistent map in large-scale environments. Usually, problems happen when a robot turns back to the starting point of its trajectory (a trajectory loop is closed). Due to accumulated errors along the trajectory, the reconstructed map is not consistent, i.e., the loop of the trajectory is not closed properly (see figs 2 and 4 in Ref. [35]). Correct data association is required to uniquely identify the landmarks corresponding to previously seen ones, from which loop-closing can be detected. Then, different techniques are applied to correct the map, for example, Kalman smoother-based and EM-based techniques.

A recent review of the approaches to the loop-closing problem can be found in Ref. [35]. From it, another advantage of EM and PF over KF can be noticed in that both EM and PF can deal with the loop-closing problem, but KF cannot. Here, we summarize two additional algorithms that are not covered. Kaess and Dellaert applied the EM approach that is employed in structure from motion. They addressed a solution by partitioning an observed point track, where a partition assigns each track to a specific structure point. If the partition is known, the motion and structure are established *a posteriori* by optimizing the likelihood over estimate of structure and motion. However, in most circumstances, partition is unknown. To solve the posterior over structure and motion given feature tracks, a Monte-Carlo EM algorithm is implemented iteratively. In the E-step, virtual structure is estimated by sampling and the virtual structure is interpreted as correct sampling divided by the total number of sampling. In the M-step, the virtual structure is employed to obtain a better motion estimate by maximizing the expected log-posterior and prior of motion.

Another important approach to vision-based loop-closing is reported by Se *et al.* [77]. Similar to the methods that globally correlate new measurements with an existing map (local registration and global correlation), the approach is to build multiple 3-D submaps which are merged together subsequently. On the basis of the distinctive features tracked by SIFT, Hough transform and RANSAC are applied for submap alignment to achieve global localization, where some overlap exists in two successive submaps. When detecting a significant overlap of landmarks between the current submap and the previously built submap, a global minimization strategy is addressed to do backward correction to all the submap alignments. By using this loop-closing constraint, the effects of accumulated error can be avoided and then a better global 3-D map is built. In the context of data association and loop-closing problems, the proposed method of combining SIFT and RANSAC provides distinctive features, which are very reliable and considered as a type of ‘fingerprint’ [77], not like other methods that commonly used only corners or lines for mapping.

5.3. Bearing only SLAM

The bearing-only SLAM problem exits in the landmark initialization of all visual SLAM solutions due to a limitation of computer vision where a meaningful range cannot be calculated from a single measurement and only the angle to landmark sightings is available. Thus, an estimate of the landmark position will not be possible until successive measurements from different points of view are made. More technically, it is not available until baseline requirements of different images have been satisfied [37]. Basically, bearing-only SLAM can be divided into two categories, i.e., delayed and undelayed initialization.

5.3.1. Delayed initialization. Due to the fact that computer vision loses one dimension of the world, 'enough baseline' requirements must be met so that past poses of the robot have to be stored, together with associated measurements. Landmark initialization is delayed until the baseline is sufficient to establish a Gaussian estimate. Two mature approaches are as follows:

- PF for detecting range proposed by Ref. [30]. The initialization is deferred until range variance is smaller than an EKF Gaussian acceptable estimate.
- Memorizing past poses with associated measurements until the requirement of the baseline is met, then calculating all delayed ranges.

5.3.2. Undelayed initialization. There are two undelayed methods reported in recent conference proceedings [36, 37]. The earlier one by Kwok and Dissanayake [36] presented a set of hypotheses for the position of the landmark and all of the hypotheses are included inside the initial map. On successive observations, the sequential ratio test (SRT) based on likelihood is used to eliminate any bad hypothesis and the one with the ML is used to correct the map.

A Federated Information Sharing-based SLAM was introduced by Solà *et al.* [37]. This technique can release the computation from Gaussian Sum Filter (GSF)-based initialization, which follows a set of weighted maps, one for each hypothesis making computational cost grow multiplicatively. GSF-SLAM proposed an additive growth of computation. The depiction can be understood in fig. 3 of Ref. [37].

Apparently, undelayed methods in bearing-only SLAM are superior to the delayed ones in at least two aspects. First, undelayed methods can avoid unnecessary storage of past poses and measurements. Second, 'enough baseline' may be difficult for some outdoor navigations when the robot travels on a straight trajectory and cameras look forward.

5.4. Kidnapped robot problem

The kidnapped robot problem describes how to recover when the robot is moved to an unknown location. Practically, there is no localization algorithm that guarantees never to fail. Therefore, it is essential for mobile robots in localization, particularly in global localization. Global landmarks and sensors, e.g., a GPS, can recover the robot from kidnapped status.

Interestingly, the solutions for the kidnapped robot can help improve SLAM algorithms. In Ref. [38], Wang *et al.* addressed a decoupled SLAM (D-SLAM) approach: a low-dimensional (robot) localization problem and static landmark mapping problem. Two methods are combined to use concurrently. One is normal SLAM and the other is the kidnapped robot solution that is given by current observation and the previously generated map. The normal SLAM solution is used unless loop-closing is present, where the kidnapped robot solution given by global localization is superior to normal SLAM under loop-closing circumstances. Such a combination offers better localization estimates than using only either one.

6. SUMMARY AND COMPARISONS

In this survey, we have reviewed the SLAM frameworks and related advances of computer vision. In this section, we will compare and summarize some important properties of these frameworks.

KF, PF and EM are probability-based techniques. They are the mathematical derivations of the recursive Bayes rule (5). The first framework we discussed is KF, which is employed by most of the visual SLAM algorithms to handle the localization and map-building uncertainties. As an advantage, KF and its extensions provide optimal MMSE estimates of the state (robot pose and landmark position). Furthermore, the covariance matrix in the KF framework is proved to converge strongly [6]. Such variance matrices (in (10) and (12)) represent the cross-correlation between landmark and robot position estimate errors, and between that of the landmarks themselves. However, the limitations of KF are also notable. For instance, its Gaussian noise assumption restricts its adaptability for data association and number of landmarks, i.e., it is only suitable for environments with a sparse set of features.

PF has some advantages over KF, such as the abilities to deal with highly nonlinear sensor and robot motion as well as non-Gaussian noise. PF is derived from SMC and Bayes rule. It implements likelihood distribution and uses particles instead of a determined point for state estimation. Due to this 'one-to-many' nature of estimation, PF has only been used for localization. Applications to PF must integrate with other techniques, e.g., EKF, to achieve the goal of SLAM.

Considering the robot performs SLAM with incomplete previous pose information, EM is a good solution. The mathematical background of EM is from ML estimation and Bayes rule. Generally, due to its characteristic of repeated data processing, EM is not suitable for incremental implementation. However, this characteristic enables the feature correspondence function for SLAM. Practical SLAM algorithms only use the M-step of the EM approach, i.e., incremental ML to map the environments, whereas localization is accomplished by other techniques.

SM-based techniques formulate the uncertainty for SLAM with the assumption of UBB instead of statistical. It models robot motion and absolute orientation noises, as well as range and bearing noises in sensor observation. All these noises, not necessary Gaussian, are bounded by some scalars so as to be embedded into measurement sets. Measurement sets integrate with the feasible set, which is the set of robot dynamics, to produce outputs for SLAM. Comparisons of these four frameworks are tabulated in Table 2. From it, the inherent properties and natural advantages or disadvantages can be understood.

Table 2.
Comparisons of the main SLAM frameworks

	KF	PF	EM/Incremental ML	SM
Probabilistic	yes	yes	yes	no
Math background	Bayesian, MMSE	SMC, Bayesian	Bayesian, ML	Set membership set
Successful application	SLAM	localization	map building, vision-based landmark representation	SLAM
Sensor noise	Gaussian	any	any	any
Incremental	yes	yes	no	yes
Uncertainty modeling	statistical	statistical	statistical	UBB assumption
Data association	no	no	yes	no

Important topics such as data association, loop-closing, bearing-only SLAM and kidnapped robot problems are either requisites of SLAM or so-called post-processing of SLAM. Increasing research interests are being drawn in these sub-disciplines of SLAM.

Current advances in camera and computer hardware give impetus for computer vision making adequate progresses in the fields of 3-D reconstruction, feature tracking and recognition, and object identification in the environment. Within the above frameworks, newly developed vision-based algorithms supply complex and nature landmark information to estimate optimal robot poses, and reconstruct accurate and high-level object maps. It can be concluded that with the development of computer vision techniques, SLAM algorithms are becoming more efficient, robust and close to a human's perception of the environment so as to facilitate the interaction between people and robots.

Acknowledgments

This work was financially supported by the National Sciences and Engineering Research Council (NSERC) of Canada, Precarn Inc. and the University of Western Ontario. Thanks also go to Dr. V. Parsa and the anonymous reviewers for helpful comments on an earlier draft of this paper.

REFERENCES

1. H. Moravec and A. Elfes, High resolution maps from wide angle sonar, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, pp. 116–121 (1985).
2. B. J. Kuipers and Y. T. Byun, A robot exploration and mapping strategy based on a semantic hierachy of spatial representations, *Int. J. Robotics Res.* **8**, 47–63 (1991).
3. E. Chown, S. Kaplan and D. Kortenkamp, Prototypes, location, and associative networks (plan): Towards a unified theory of cognitive mapping, *Cognitive Sci.* **19**, 1–51 (1995).
4. R. Smith and P. Cheeseman, On the representation of spatial uncertainty, *Int. J. Robotics Res.* **5**, 56–68 (1986).

5. R. Smith, M. Self and P. Cheeseman, Esimating uncertain spatial relationships in robotics, in: *Autonomous Robot Vehicles* (I. J. Cox and G. T. Wilfong, Eds), pp. 167–198. Springer, New York, NY (1990).
6. M. W. M. G. Dissanayake, P. Newman and S. Clark, A solution to the simultaneous localizaiton and map building (SLAM) problem, *IEEE Trans. Robotics Automat.* **17**, 229–241 (2001).
7. J. Kim and S. Sukkarieh, Autonomous airborne navigation in unknown terrain environments, *IEEE Trans. Aerospace Electron. Syst.* **40**, 1031–1045.
8. A. J. Davison and D. W. Murray, Simultaneous localization and map-building using active vision, *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 865–880 (2002).
9. J. Castellanos, J. Neira and J. Tardos, Multisensor fusion for simultaneous localization and map building, *IEEE Trans. Robotics Automat.* **17**, 908–914 (2001).
10. N. Tomatis, I. Nourbakhsh and R. Siegwart, Hybrid simultaneous localization and map building: a natural integration of topological and metric, *Robotics Autonomous Syst.* **44**, 3–14 (2003).
11. R. Madhavan, K. Fregene and L. E. Parker, Distributed cooperative outdoor multirobot localization and mapping, *Autonomous Robots* **17**, 23–39 (2004).
12. E. Hygounenc, I. K. Jung, P. Soueres and S. Lacroix, The autonomous blimp project of LAAS-CNRS: achievements in flight control and terrain mapping, *Int. J. Robotics Res.* **23** 00–00 (2004).
13. S. Se, D. Lowe and J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks, *Int. J. Robotics Res.* **21**, 735–758 (2002).
14. R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. P. Ostrowski, G. Pappas, C. J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic and B. Southall, A framework and architecture for multi-robot coordination, *Int. J. Robotics Res.* **21**, 977–995 (2002).
15. V. A. Sujan and S. Dubowsky, Efficient information-based visual robotic mapping in unstructured environments, *Int. J. Robotics Res.* **24**, 275–293 (2005).
16. V. A. Sujan, S. Dubowsky, T. Huntsberger, H. Aghazarian, Y. Cheng and P. Schenker, An architecture for distributed environment sensing with application to robotic cliff exploration, *Autonomous Robots* **16**, 287–311 (2004).
17. J. Porta, J. Verbeek and B. Krose, Active appearance-based robot localization using stereo vision, *Autonomous Robots* **18**, 59–80 (2005).
18. P. Pirjanian, N. Karlsson, L. Goncalves and E. D. Bernardo, Low-cost visual localization and mapping for consumer robotics, *Industrial Robot* **30**, 139–144 (2003).
19. M. Jogan, M. Artac, D. Skocaj and A. Leonardis, A framework for robust and incremental self-localization of a mobile, *Computer Vision Syst.* (*LNCS*), 460–469 (2003).
20. S. Thrun, D. Fox and W. Burgard, A probabilistic approach to concurrent mapping and localization for mobile robots, *Machine Learn.* **31**, 29–53 (1998); also appeared in *Autonomous Robots* **5**, 253–271 (joint issue).
21. M. Di-Marco, A. Garulli, S. Lacroix and A. Vicino, Set membership localization and mapping for autonomous navigation, *Int. J. Robust Nonlinear Control* **11**, 709–734 (2001).
22. M. Di-Marco, A. Garulli, A. Giannitrapani and A. Vicino, Simultaneous localization and map building for a team of cooperating robots: a set membership approach, *IEEE Trans. Robotics Automat.* **19**, 238–249 (2003).
23. P. Newman, On the structure and solution of the simultaneous localization and map building problem, *PhD Dissertation*, Australian Centre for Field Robotics, University of Sydney (1999).
24. J. M. Sáez, A. Penalver and F. Escolano, Compact mapping in plane-parallel environments using stereo vision, *Progr. Pattern Recognit. Speech Image Anal.* (*LNCS*) **2905**, 659–666 (2003).
25. S. Thrun and Y. Liu, Multi-robot SLAM with sparse extended information filers, in: *Proc. 11th Int. Symp. of Robotics Research*, Sienna, pp. 1–12 (2003).
26. L. Kleeman and R. Kuc, Mobile robot sonar for target localization and classification, *Int. J. Robotics Res.* **14**, 295–318 (1995).

27. L. Kleeman, Real time mobile robot sonar with interference rejection, *Sonar Rev.* **19**, 214–221 (1999).
28. S. Thrun, *Robotic mapping: a survey*, in: *Exploring Artificial Intelligence in the New Millenium* (G. Lakemeyer and B. Nebel, Eds), pp. 1–35. Morgan Kaufmann, San Mateo, CA (2002).
29. Y. Ma, S. Soatto, J. Kosecka and S. S. Sastry, *An Invitation to 3-D Vision*. Springer-Verlag, New York (2004).
30. A. Davison, Real-time simultaneous localisation and mapping with a single camera, in: *Proc. Int. Conf. on Computer Vision*, Nice, pp. 1403–1416 (2003).
31. M. Montemerlo, *FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association*. Ph. D. Thesis, Carnegie Mellon University, Pittsburgh, PA (2003).
32. D. Wolter, L. Latecki, R. Lakamper and X. Sun, Shape-based robot mapping, *Adv. Artif. Intell.* (*LNCS*) **3238**, 439–452 (2004).
33. G. N. DeSouza and A. C. Kak, Vision for mobile robot navigation: a survey, *IEEE Trans. Pattern Anal. and Mach. Intell.* **24**, 237–267 (2002).
34. M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit, FastSLAM: A factored solution to the simultaneous localization and mapping problem, in: *Proc. AAAI Nat. Conf. on Artificial Intelligence*, Edmonton, pp. 593–598 (2002).
35. M. Kaess and F. Dellaert, A Markov chain Monte Carlo approach to closing the loop in: SLAM, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Barcelona, pp. 643–648 (2005).
36. N. W. Kwok and G. Dissanayake, An efficient multiple hypothesis filter for bearing-only SLAM, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, pp. 736–741 (2004).
37. J. Solà, A. Monin, M. Devy and T. Lemaire, Undelayed initialization in bearing only SLAM, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, pp. 2499–2504 (2005).
38. Z. Wang, S. Huang and G. Dissanayake, Decoupling localization and mapping in SLAM using compact relative maps, in: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, Edmonton, AB, August 2–6, pp. 3336–3341 (2005).
39. Y. Bar-Shalom, X.-R. Li and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley, New York (2001).
40. S. Hayin, *Adaptive Filter Theory*, 4th edn. Prentice-Hall, Upper Saddle River, NJ (2002).
41. P. A. Beardsley, I. D. Reid, A. Zisserman and D. W. Murray, Active visual navigation using non-metric structure, in: *Proc. 5th Int. Conf. on Computer Vision*, Cambridge, MA, pp. 58–64 (1995).
42. A. J. Davison, Mobile robot navigation using active vision, *PhD Dissertation*, University of Oxford (1998).
43. B. R. Fajen, W. H. Warren, S. Temizer and L. P. Kaelbling, A dynamical model of visually-guided steering, obstacle avoidance, and route selection, *Int. J. Comp. Vision* **54**, 13–34 (2003).
44. J. Santos-Victor, G. Sandini, F. Curotto and S. Garibaldi, Divergent stereo in autonomous navigation: From bees to robots, *Int. J. Comp. Vision* **14**, 159–177 (1995).
45. M. C. Andrea Giachetti and V. Torre, The use of optical flow for road navigation, *IEEE Trans. Robotics Automat.* **14**, 34–48 (1998).
46. R. Lerner, E. Rivlin and P. Rotstein, Error analysis for a navigation algorithm based on optical-flow and a digital terrain map, in: *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Washington, DC, vol. 1 pp. 604–610 (2004).
47. C. Thorpe, M. H. Hebert, T. Kanade and S. A. Shafer, Vision and navigation for the carnegie-mellon navlab, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**, 362–373 (1988).
48. M. A. Sotelo, F. J. Rodriguez and L. Magdalena, Virtuous: vision-based road transportation for unmanned operation on urban-like scenarios, *IEEE Trans. Intell. Transport. Syst.* **5**, 69–83 (2004).
49. J. Lobo and J. Dias, Vision and inertial sensor cooperation using gravity as a vertical reference, *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 1597–1608 (2003).

50. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd edn. Prentice-Hall, Upper Saddle River, NJ (2002).
51. A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ (1986).
52. B. K. P. Horn, *Robot Vision*. MIT Press, Cambridge, MA (1986).
53. D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ (1982).
54. O. Faugeras, *Three-Dimensional Computer Vision: A Gerometric Viewpoint*. MIT Press, Cambridge, MA (1993).
55. Y.-S. Yao and R. Chellappa, Tracking a dynamic set of feature points, *IEEE Trans. Image Process.* **4**, 1382–1395 (1995).
56. T. Gevers, Robust segmentation and tracking of colored objects in video, *IEEE Trans. Circuits Syst. Video Technol.* **14**, 776–781 (2004).
57. C. S. Kenney, M. Zuliani and B. S. Manjunath, An axiomatic approach to corner detection, in: *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, vol. 1, pp. 191–197 (2005).
58. D. G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision* **60**, 31–110 (2004).
59. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd edn. Cambridge University Press, Cambridge (2003).
60. O. Faugeras and Q.-T. Long, *The Geometry of Multiple Images*. MIT Press, Cambridge, MA (2001).
61. Y. Lu, J. Z. Zhang, Q. M. J. Wu and Z.-N. Li, A survey of motion-parallax-based 3-D reconstruction algorithms, *IEEE Trans. Syst. Man Cybernet.* **99**, 532–548 (2004).
62. A. Fusiello, Uncalibrated Euclidean reconstruction: a review, *J. Image and Vision Comput.* **18**, 555–563 (2000).
63. Z. Zhang, Determining the epipolar geometry and its uncertainity: a review, *Int. J. Comp. Vision* **27**, 161–195 (1998).
64. C. Tomasi and T. Kanade, Shape and motion from image streams under orthography: a factorization method, *Int. J. Comp. Vision* **9**, 137–154 (1992).
65. B. Triggs, Factorization methods for projective structure and motion, in: *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 845–851 (1996).
66. M. Han and T. Kanade, Multiple motion scene reconstruction with uncalibrated cameras, *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 884–894 (2003).
67. M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops and R. Koch, Visual modeling with a hand-held camera, *Int. J. Comp. Vision* **59**, 207–232 (2004).
68. M. Han and T. Kanade, A perspective factorization method for Euclidean reconstruction with uncalibrated cameras, *J. Visualizat. Comp. Animat.* **13**, 211–223 (2002).
69. M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using Matlab*. Wiley, New York (2002).
70. Y. C. Ho and R. C. K. Lee, A Bayesian approach to problems in stochastic estimation and control, *IEEE Trans. Automat. Control* **AC-9**, 333–339 (1964).
71. P. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 1. Academic Press, New York (1979).
72. A. Davison and D. Murray, Mobile robot localization using active vision, in: *Proc. of 5th Eur. Conf. on Computer Vision*, Freiburg, pp. 809–825 (1998).
73. F. Marchand and E. Chaumette, An autonomous active vision system for complete and accurate 3-D scene reconstruction, *Int. J. Comp. Vision* **32** 171–194 (1999).
74. J. A. Castellanos, Mobile robot localization and map building, *PhD Dissertation*, University of Zaragoza (1999).

75. N. Tomatis, I. Nourbakhsh, K. Arras and R. Siegwart, A hybrid approach for robust and precise mobile robot navigation with compact environment modeling, in: *Proc. IEEE Int. Conf. on Robotics and Automation* Seoul, pp. 2051–2058 (2001).
76. S. Se, D. Lowe and J. Little, Local and global localization for mobile robots using visual landmarks, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Maui, HI, pp. 414–420 (2001).
77. S. Se, D. G. Lowe and J. J. Little, Vision-based global localization and mapping for mobile robots, *IEEE Trans. Robotics* **21**, 364–375 (2005).
78. J. Kim and S. Sukkarieh, Airborne simultaneous localisation and map building, in *Proc. IEEE Int. Conf. on Robotics and Automation*, Taipei, pp. 406–411 (2003).
79. J. Kim, S. Sukkarieh and S. Wishart, Real-time navigation, guidance and control of a UAV using low-cost sensors, in: *Proc. Int. Conf. on Field and Service Robotics*, Yamanashi, pp. 95–100 (2003).
80. P. G. Savage, Strapdown inertial navigation integration algorithm design. Part I: attitude algorithms. *J. Dyn. Syst. Meas. Control* **21**, 19–28 (1998).
81. P. G. Savage, Strapdown inertial navigation integration algorithm design. Part II: velocity and position algorithms, *J. Dyn. Syst. Meas. Control* **21**, 208–221 (1998).
82. A. Davison, Y. G. Cid and N. Kita, Real-time 3-D SLAM with wide-angle vision, in: *Proc. IFAC Symp. on Intelligent Autonomous Vehicles*, Lisbon (2004). Available online at: http://www.doc.ic.ac.uk/∼ajd/Publications/davison_etal_iav2004.pdf.
83. Z. Chen and J. Samarabandu, Using multiple view geometry within extended Kalman filter framework for simultaneous localization and map-building, in: *Proc. IEEE Int. Conf. on Mechatronics and Automation*, Niagara Falls, pp. 695–700 (2005).
84. S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte and N. A. Y., Simultaneous mapping and localization with sparse extended information filters, in: *Proc. 5th Int. Workshop on Algorithmic Foundations of Robotics*, Nice, pp. 363–380 (2002).
85. S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani and H. Durrant-Whyte, Simultaneous localization and mapping with sparse extended information filters, *Int. J. Robotics Res.* **23**, 693–716 (2004).
86. B. Ristic, S. Arulampalam and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, MA (2004).
87. D. C. Yuen and B. A. MacDonald, An evaluation of the sequential Monte Carlo technique for simultaneous localisation and map-building, in: *Proc. Int. Conf. on Robotics and Automation*, Taipei, pp. 1564–1569 (2003).
88. K. Murphy, Bayesian map learning in dynamic environments, *Adv. Neural Inform. Process. Syst.* **12**, 1015–1021 (2000).
89. B. Terwijn, J. Porta and B. Kröse, A particle filter to estimate non-Markovian states, in: *Proc. 8th Int. Conf. on Intelligent Autonomous Systems*, Amsterdam, pp. 1062–1069 (2004).
90. R. Sim and G. Dudek, Comparing image-based localization methods, in *Proc. 18th Int. Joint Conf. on Artificial Intelligence*, Acapulco, pp. 1560–1562 (2003).
91. J. E. Banta, L. Wong, C. Dumont and M. Abidi, A next-best-view system for autonomous 3-D object reconstruction, *IEEE Trans. Syst. Man Cybernet. A* **30**, 589–598 (2000).
92. G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Wiley, New York (1997).
93. S. Thrun, A probabilistic online mapping algorithm for teams of mobile robots, *Int. J. Robotics Res.* **20**, 335–363 (2001).
94. M. Jogan and A. Leonardis, Robust localization using eigenspace of spinning-images, in: *Proc. IEEE Workshop on Omnidirectional Vision*, Hilton Head Island, SC, pp. 37–44 (2000).
95. Y. Liu, R. Emery, D. Chakrabarti, W. Burgard and S. Thrun, Using EM to learn 3-D models of indoor environments with mobile robots, in: *Proc. Int. Conf. on Machine Learning*, Williamstown, MA, pp. 329–336 (2001).

96. F. Dellaert, S. Seitz, C. Thorpe and S. Thrun, EM, MCMC, and chain flipping for structure from motion with unknown correspondence, *Machine Learn.* **50**, 1–2 (2003).
97. M. Milanese, J. P. Norton, H. Piet-Lahanier and E. Walter, *Bounding Approaches to System Identification*. Plenum Press, New York (1996).
98. D. Hähnel, W. Burgard, B. Wegbreit and S. Thrun, Towards lazy data association in SLAM, in: *Proc. 11th Int. Symp. of Robotics Research*, Sienna, pp. 421–431 (2003).
99. J. Neira and J. D. Tardos, Data association in stochastic mapping using the joint compatibility test, *IEEE Trans. Robotics Automat.* **17**, 890–897 (2001).

ABOUT THE AUTHORS

Jagath Samarabandu received his BS in Electronics and Telecommunication with first class honours from the University of Moratuwa, Sri Lanka in 1982. He was awarded a Fulbirght Scholarship in 1987 for postgraduate study. He received the MS and PhD degrees in Electrical Engineering from State University of New York (SUNY) at Buffalo, in 1990 and 1994, respectively. He held a post-doctoral position in the Department of Biological Sciences at SUNY at Buffalo untill 1997 and joined Life Imaging Systems Inc. as a Senior Software Engineer in 1997. He has been with the Department of Electrical and Computer Engineering at the University of Western Ontario since 2000. His research interests include image analysis, pattern recognition and intelligent systems.

Ranga Rodrigo obtained his BS degree from the University of Moratuwa, Sri Lanka, with first class honors, in 2000. After working there at the same university as a Probationary Lecturer, he joined the University of Western Ontario in 2003. There he obtained his MS degree in 2005. Currently, he is reading for his PhD degree at the same university. His research interests are in the general area of computer vision including three-dimensional reconstruction and navigation. He has been a reviewer for conferences in his area.

Zhenhe Chen received the BE degree in Electrical Engineering from South China University of Technology, Guangzhou, China, in 1996, and the MAS degree in Electrical and Computer Engineering from Ryerson University, Toronto, Canada, in 2003. He is presently completing his PhD degree in Electrical and Computer Engineering at the University of Western Ontario. From 1996 to 2000 he worked for State China Administration of Taxation as a Project Coordinator of the Golden Tax of the Chinese Government. He is a recipient of Precarn Scholarship in 2006. His research interests include robot navigation within probabilistic frameworks, computer vision and video segmentation by independent component analysis. He has been a reviewer for conferences in his area of research.