

Department of Electrical and Computer Systems Engineering

Technical Report MECSE-27-2005

A New Solution to the Simultaneous Localisation and Map
Building (SLAM) Problem

D. J. Spero and R. A. Jarvis

MONASH
UNIVERSITY

A New Solution to the Simultaneous Localisation and Map Building (SLAM) Problem

Dorian J. Spero and Ray A. Jarvis
Intelligent Robotics Research Centre
Monash University,
Clayton, Victoria 3800, Australia
Email: dorian@ieee.org

Abstract—The proposition posed by the problem of simultaneous localisation and map building (SLAM) is whether a mobile robot can be placed in an unknown environment and then incrementally build a map of this environment, while using the map to determine its globally referenced pose. The various solutions to the SLAM problem that have been proposed so far, such as the extended Kalman filter (EKF), are typically highly theoretical and require rigorous modeling of the robot's locomotion mechanism, sensor errors and the environment. These motifs have encouraged a long string of simplifying and often wishful assumptions, which invariably restrict the application of SLAM to only a contrived figment of the real world. While using a plethora of assumptions may be academically acceptable, a change of tack is needed for real-world operation.

In this paper, a novel SLAM solution is proposed that is based on perpetually solving the kidnapped robot problem. By doing so, the locomotive specifics of the robot are irrelevant and, hence, the robot can be engineered to be as flexible and robust as practicable without contemplating the possible side effects on odometric accuracy and associated measurement drift. While the proposed solution does not require odometry (nor does it assume motion continuity), it makes provisions for the inclusion of such information, in possible combination with inertial sensors, to improve the system's performance while not compromising its generality.

The solution comprises a multiple-hypothesis data association algorithm for recognising map landmarks perceived from different viewpoints, and a simple relative error algorithm for representing and handling the positional uncertainties of the robot and landmarks. The latter algorithm is based on the sole premise that the positional uncertainty is directly proportional to the radial distance from the origin. This paper argues that the proposed solution, while atypical, facilitates navigation in natural environments. Practical results from several outdoor experimental trials will be published shortly.

Index Terms—Autonomous robot navigation; SLAM; localisation; environment mapping; kidnapped robot problem

I. INTRODUCTION

Simultaneous localisation and map building (SLAM) is the process of concurrently building a landmark based map of the environment and using this map to ascertain the robot's absolute pose. Initially, the robot starts at an unknown location in an *a priori* unknown environment. It then uses its onboard sensors to observe the local landmarks, and from this information, computes its own pose while simultaneously estimating the locations of these landmarks. As the robot moves through the environment, its changing

observational viewpoint enables the incremental building of a complete map of landmarks, which are continuously exploited to track the robot's current pose relative to its initial pose.

In mathematical terms, the objective of SLAM is to estimate the system state \mathbf{x}_k at discrete time instant k , given by

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{r_k} \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} \quad (1)$$

where \mathbf{x}_{r_k} is the robot's state and the set $M = \{\mathbf{x}_i | 1 \leq i \leq n\}$ represents the map of observed landmarks. Notice that the landmark states \mathbf{x}_i are not given as a function of time, as they are generally assumed to be stationary. Moving environmental features are consequently disregarded as unwanted noise, however, they can still be useful to the SLAM process, especially if their movement is predictable, intermittent, or negligible within a sparse environment.

For a 2D Cartesian based map, the robot's state can be defined by its pose (position and orientation) in space

$$\mathbf{x}_{r_k} = \begin{bmatrix} x_{r_k} \\ y_{r_k} \\ \theta_{r_k} \end{bmatrix} \quad (2)$$

relative to a global reference frame, as shown in Fig. 1. The landmarks in the map M are commonly represented as points in space and therefore their states may be defined by

$$\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \quad (3)$$

However, the equations (2) and (3) vary according to the robot's intended application and the particular SLAM strategy used.

Now that a definition of SLAM has been given, what makes the practical operation of SLAM a problem? The answer to this question, as discussed in [1], lies in the difficulty of coping with three distinct forms of uncertainty:

- Data association uncertainty
- Navigation error
- Sensor noise

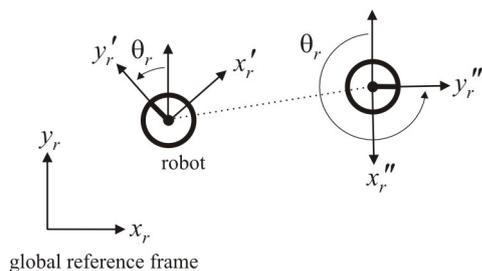


Fig. 1
ROBOT COORDINATE SYSTEM

Data association uncertainty occurs due to the robot's inability to properly identify a landmark perceived from different poses as the same. Therefore there exists the possibility that the robot wrongly associates landmarks, thereby corrupting the map. This problem is generally referred to as either the *data association problem* or *correspondence problem* [2, 3]. The second form of uncertainty, i.e., navigation error, is caused by the inevitable divergence between the robot's assumed motion via the vehicle model and its actual motion. This divergence can lead to an accumulative error in the robot's estimated pose as well as exacerbate the data association problem. The final form of uncertainty is the result of imperfect sensing devices, the measurements of which are inherently noisy and sometimes completely erroneous.

Together, these uncertainties culminate into a complex SLAM problem, one which some behaviourists believe is not worth solving [4]. Their argument is that humans and animals can navigate perfectly well without precise quantitative knowledge of their location, so why should their mechanical counterparts need to perform SLAM? A rebuttal to this argument is that there are a range of useful applications such as cross-country and interplanetary exploration, undersea navigation and mining where the robot needs to track its precise position over a long-term period without the aid of an *a priori* map or artificial infrastructure [5]. By using SLAM, the robot is able to navigate efficiently and purposefully within its *a priori* unknown environment, while strategically carrying out its mission. In fact, some researchers go beyond this conservative view by stiling the solution to the SLAM problem as the cornerstone or "Holy Grail" of robot autonomy [6]. For these reasons, the SLAM problem has received a considerable amount of research attention, and judging by its dominant discourse at international conferences, the number of active researchers in this area is growing rapidly.

A number of methods have been proposed to solve the SLAM problem, each with relative strengths and practical limitations. Section II provides a review of these methods and delves into some of the more notable instances, such as the *extended Kalman filter (EKF)* and *particle filter* based approaches, for a look into their inner workings. From this review, it is apparent that the predominant downside

of these methods is their reliance on stringent models of the robot, its sensors, and the environment (i.e., the supporting medium through which the robot locomotes and the observable features). The roboticist (or engineer) implementing one of these methods then has the dubious task of modeling these aspects, while appropriately constraining the aforementioned uncertainties so that the model boundaries remain intact. The intractable real-world, however, is not amenable to such artificial boundaries; and the more rigid the models, the tighter these boundaries become. So, to achieve some semblance of reliability, it has become common practice to manufacture the robot and its operating world in such a way that they are submissive to the modeling process. Therefore the models entice contrivances which are detrimental to real-world operation.

In Section III, a new SLAM solution is proposed that moves away from the classical ideal of modeling and pre-planning for the purpose of gaining system generality and, consequently, robustness and flexibility when deployed in the real-world. This solution is based on perpetually solving the *kidnapped robot problem* [7], which is defined as the problem of re-localising a mobile robot after its undergone an unknown motion or, in figurative terms, been kidnapped and clandestinely placed at an unknown location. The kidnapped robot problem is typically considered in the context of a one-off, unwanted navigation event that needs to be first detected and then resolved. However, in this case, it is assumed to be continuously occurring over time, and by perpetually solving such a problem, the specifics of the robot's locomotion mechanism becomes irrelevant. So, whether the robot locomotes via wheels, tracks, limbs, or one of the recent self-reconfiguring designs [8, 9, 10], has no bearing on the working operation of the proposed SLAM solution.

Several other important outcomes are also realised. Firstly, odometry and an associated vehicle model are not required, and so this solution is not vulnerable to the large *non-systematic errors* [11] which can occur from the physical interaction between the robot and its environment (e.g., outdoor surface irregularities causing a robot's wheels to slip or unpredictable undercurrents acting on a submersible). Another important outcome is that there is no assumption of continuity in the robot's motion, i.e., \mathbf{x}_{r_k} is not constrained by $\mathbf{x}_{r_{k-1}}$. This means that the SLAM process is essentially decoupled from the robot itself and, ergo, has a similar flexibility to the Global Positioning System (GPS) in terms of a standalone black-box device. The difference being that instead of communicating with orbiting satellites like a GPS receiver, the proposed solution functions by sensing the local environment.

However, purposely disregarding odometry and the constraints imposed by the robot's locomotive mechanism is an extreme approach to SLAM that may seem illogical, especially if this information is readily available. In rebuttal to this, the proposed solution only disregards this information in lure of its generalised applicability to an arbitrary robot, regardless of the robot's suitability to modeling. But in a

context specific situation, however, provisions have been made for the inclusion of any available dead-reckoning information from odometry and possibly inertial sensors (gyroscopes and accelerometers) in the SLAM process. This information can be added at will, with varying degrees of accuracy, to improve the system's performance characteristics without the repercussive effect of it becoming a point of vulnerability. That is, regardless of whether dead-reckoning information is added, the proposed solution is still fundamentally based on solving the kidnapped robot problem and so it is only vulnerable to its authoritative reference – the environment – and not the robot itself.

The proposed solution consists of two subsystems: *multiple-hypothesis data association* and *positional error representation and handling*. While the solution from a holistic perspective is novel in itself, these subsystems are also novel in their approach and methods. The multiple-hypothesis data association subsystem is used to match landmarks perceived from different robot locations and, correspondingly, address any uncertainty or ambiguity that results from the matching process. The other subsystem is used to maintain the positional errors of the robot and landmarks. Both subsystems are presented in detail.

Finally, Section IV concludes the paper with some of the many directions for future research. Since the presented solution is original, it lacks the years of development afforded to the classical approaches, like the EKF, and so numerous extensions are possible as well as rigorous comparative studies and analyses. It is argued that the solution's generality warrants such investigation.

II. RELATED WORK

This section reviews the current state of the art in solving the SLAM problem. At present, there are too many approaches to solving this problem, and too many extensions and subtleties within each approach, to review all of them in any great detail here; so only the more notable instances will be considered. While we acknowledge that this form of selectivity has a subjective connotation, the aim of this review is to just give a well-rounded overview of the area and highlight some of the current limitations that exist, especially with regard to real-world operation.

The various approaches will be compared on the basis of several key properties. These properties may include, for example, the map representation (e.g. Cartesian landmark locations, occupancy grid, or polygons); the representation of uncertainty in the map (e.g. Gaussian / mixture of Gaussians, maximum likelihood, or a particle set); restrictions on sensor noise; optimality of convergence; computational complexity; accuracy (with respect to ground truth); generality / applicability; robustness to unmodeled events (e.g. large non-systematic motion errors, dynamic environmental features, or incorrect data association); and whether the map is incrementally built, and if so, the consistency of the resultant map.

A property that is rarely, if ever, mentioned in the literature is the required academic competence of the roboticist

implementing the approach. Since highly theoretical approaches can be complicated and time-consuming to implement by roboticists who are not *au fait* with the underlying principles involved, this property will be discussed here under the umbrella of "ease of implementation". Another factor affecting the ease of implementation is the amount of preparatory analysis required to fulfill an approach's prerequisites (e.g., model parameters). Approaches that are relatively easy to implement with little preparatory work will be considered better than those that lean towards the opposite way, all else being equal. This may seem obvious; however, complicated approaches can offer their inventors the power of exclusivity, which will not be considered an advantage here.

The review begins with what is currently the most popular approach: the EKF. This approach will be used as a benchmark for comparison between the other approaches.

A. The Estimation-Theoretic Approach

The *estimation-theoretic* or *extended Kalman filter (EKF) based approach* was first introduced by Smith, Self and Cheeseman in their seminal paper [12], which described the use of an EKF [13] to build a stochastic map of spatial relationships. This work was extended shortly after by Moutarlier and Chatila [14], who took into account the correlated noise between landmarks in the map to preserve the filter's consistency. Leonard and Durrant-Whyte [15]¹ then implemented it using an indoor mobile robot equipped with sonar sensors. Since then, a considerable amount of progress has been made in the development of the EKF based approach, including such contributions as its application to different domains [16, 17]; the use of various sensors [18, 19]; proofs of its convergence properties [6] or lack thereof [20]; and methods that somewhat address its high computational complexity of $\mathcal{O}(n^2)$ [21, 22, 23, 24, 25].

The mathematical framework of the EKF is based on a state space representation of the robot and its environment. In presenting the mathematical framework here, the system state vector \mathbf{x}_k given in Section I will be used and several models will be introduced. The first model, called the system plant model, describes how the system states change as a function of time k and is conventionally written as a non-linear state transition equation of the form

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{v}_k \quad (4)$$

where \mathbf{u}_k represents the control input asserted in the time interval $(t_{k-1}, t_k]$, \mathbf{v}_k denotes temporally uncorrelated Gaussian noise with zero mean ($E[\mathbf{v}_k] = 0, \forall k$) and covariance \mathbf{Q}_k , and $\mathbf{f}(\cdot, \cdot)$ is a non-linear function that maps \mathbf{x}_{k-1} to \mathbf{x}_k given \mathbf{u}_k . Similarly, a robot or vehicle model is used to capture the robot's progression from its previous state, $\mathbf{x}_{r_{k-1}}$, to the next, \mathbf{x}_{r_k} , as determined by its kinematics, and can be written as

$$\mathbf{x}_{r_k} = \mathbf{f}_r(\mathbf{x}_{r_{k-1}}, \mathbf{u}_{r_k}) + \mathbf{v}_{r_k}. \quad (5)$$

¹The origin of the phrase "simultaneous localisation and map building".

Assuming that the landmarks in the map M are stationary, the landmark model is trivially

$$\mathbf{x}_{i,k} = \mathbf{x}_{i,k-1} \quad (6)$$

and therefore the dynamics of the system is confined to the robot model. During the robot's motion, it uses an onboard sensor, or a multisensor arrangement [26], to observe the local landmarks and measure their relative position. This is represented by an observation model where the observation at time k , denoted \mathbf{z}_k , is expressed in the form

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k \quad (7)$$

where \mathbf{w}_k is a random vector of temporally uncorrelated measurement noise with zero mean ($E[\mathbf{w}_k] = 0, \forall k$) and covariance \mathbf{R}_k , and $\mathbf{h}(\cdot)$ is a non-linear function that models the relationship between the observation of system states and the states themselves.

Based on the system and observation models given in (4) and (7), respectively, the EKF fuses all the available information about the system's state to compute a state estimate with minimum mean-squared error (MMSE). This is accomplished through a recursive, three-stage cycle consisting of *prediction*, *observation*, and *update* steps [27].

Since the EKF equations [13, 28] that make up these steps have been widely published with many notational nuances, the notation used to present them here will be briefly described first. The notation $\hat{\mathbf{x}}_k^-$ represents the *a priori* state estimate at time k or, in other words, the state prediction derived from information up to time $k-1$ (i.e., $\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k|k-1}$). Conversely, $\hat{\mathbf{x}}_k^+$ represents the *a posteriori* state estimate at time k and therefore is conditioned on information up to this time (i.e., $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_{k|k}$). Note that the '+' and '-' superscripts will also be used for other state variables to convey the same meaning.

1) *Prediction*: The first step of the filter involves generating predictions of the system's state $\hat{\mathbf{x}}_k^-$, its covariance \mathbf{P}_k^- , and the observation $\hat{\mathbf{z}}_k^-$ at time k . These predictions are calculated as follows:

$$\hat{\mathbf{x}}_k^- = \mathbf{f}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k) \quad (8)$$

$$\hat{\mathbf{z}}_k^- = \mathbf{h}(\hat{\mathbf{x}}_k^-) \quad (9)$$

$$\mathbf{P}_k^- = \nabla \mathbf{f}_{\mathbf{x}_{k-1}} \mathbf{P}_{k-1}^+ \nabla \mathbf{f}_{\mathbf{x}_{k-1}}^T + \mathbf{Q}_k \quad (10)$$

$$\text{where } \nabla \mathbf{f}_{\mathbf{x}_{k-1}} \triangleq \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k)} \quad (11)$$

The Jacobian $\nabla \mathbf{f}_{\mathbf{x}_{k-1}}$, defined in equation (11), is derived from linearising the non-linear function \mathbf{f} through a first-order Taylor series expansion about the point of $\hat{\mathbf{x}}_{k-1}^+$. Also, note that equation (10) does not take into account the uncertainty in the control inputs \mathbf{u}_k , however, this can be remedied by adding the term $\nabla \mathbf{f}_{\mathbf{u}_k} \mathbf{U}_k \nabla \mathbf{f}_{\mathbf{u}_k}^T$ (where \mathbf{U}_k is the control covariance) to the right side of this equation.

2) *Observation*: After the robot makes a partial observation \mathbf{z}_k of the true landmark states in \mathbf{x}_k , the innovation ν_k is calculated using

$$\nu_k = \mathbf{z}_k - \hat{\mathbf{z}}_k^- \quad (12)$$

under the assumption of perfect data association. The corresponding innovation covariance \mathbf{S}_k is calculated as follows:

$$\mathbf{S}_k = \nabla \mathbf{h}_{\mathbf{x}_k} \mathbf{P}_k^- \nabla \mathbf{h}_{\mathbf{x}_k}^T + \mathbf{R}_k \quad (13)$$

$$\text{where } \nabla \mathbf{h}_{\mathbf{x}_k} \triangleq \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-} \quad (14)$$

Similar to the Jacobian $\nabla \mathbf{f}_{\mathbf{x}_{k-1}}$ described earlier, the Jacobian $\nabla \mathbf{h}_{\mathbf{x}_k}$ is a linearisation of the observation function \mathbf{h} .

3) *Update*: The final step involves updating the state estimate $\hat{\mathbf{x}}_k^+$ and its covariance \mathbf{P}_k^+ according to the following equations:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{W}_k \nu_k \quad (15)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{W}_k \mathbf{S}_k \mathbf{W}_k^T \quad (16)$$

where the Kalman gain \mathbf{W}_k is given by

$$\mathbf{W}_k = \mathbf{P}_k^- \nabla \mathbf{h}_{\mathbf{x}_k}^T \mathbf{S}_k^{-1} \quad (17)$$

Overall, this filter provides a theoretically sound solution to SLAM and a means of systematically studying its convergence properties, the evolution of the map, and the propagation of positional uncertainties. However, from a practical standpoint, there are several issues that adversely affect its applicability. To begin with, the approximation errors caused by linearising the system and measurement functions can lead to filter instability and an inconsistent map [20, 29], especially if the time step interval Δt_k (where $\Delta t_k = t_k - t_{k-1}$) is not sufficiently small. Julier and Uhlmann partially solved this problem by introducing the unscented Kalman filter (UKF) [30], which tends to be more suited to highly non-linear functions than the EKF. However, both of these extensions to the standard Kalman filter are still limited by their inherent assumptions, such as Gaussianity and independence of model errors, which realistically may not hold true.

Another limitation of the EKF is that landmarks need to be uniquely identified by the data association process. For instance, it is not enough to just be able to recognise that a certain percept is a tree, the tree has to be matched to its corresponding landmark state in the map. Since data association is commonly performed using the *gated nearest-neighbour (NN) algorithm* [2], this type of identification becomes increasingly more unreliable as environmental clutter or uncertainty in the robot's estimated state $\hat{\mathbf{x}}_k^-$ grows. This can cause false data associations,

which then often lead to catastrophic failure [27]. However, the likelihood of this happening can be reduced by applying a more robust data association technique such as the *joint compatibility test* [31] or the *graph theoretic approach* [32]. Ambiguous observation data can also be better handled using *multiple hypothesis tracking (MHT)* [33, 34], which maintains each possible interpretation of the data over time using multiple, probabilistically weighted, maps. These enhancements, however, add to the computational complexity of the EKF.

Lastly, the biggest problem with the EKF is arguably its reliance on stringent models to satisfy its predictive behaviour. This reliance means that the operational performance of the EKF is extremely specific to the extent of which the robot and its environment are predisposed to the modeling process. Consequently, the robot designs and environments that cannot be easily modeled or manipulated are often avoided, and those that can are tightly bounded with little tolerance for the unknown. There are other probabilistic approaches to SLAM that are not as rigid.

B. Other Probabilistic Approaches

To begin to describe these other approaches, it is worthwhile to look at the SLAM problem from a probabilistic point of view. The SLAM problem in this context is considered to be a density estimation problem where the solution involves finding the joint posterior probability of the robot's pose \mathbf{x}_{r_k} and map M at time k . This posterior can be written as

$$p(\mathbf{x}_{r_k}, M | \mathbf{z}_{0:k}, \mathbf{u}_{0:k}) \quad (18)$$

where $\mathbf{z}_{0:k}$ and $\mathbf{u}_{0:k}$ represent the observation and control history, respectively. For notational convenience, this posterior will be denoted $b_k(\mathbf{x}_{r_k}, M)$ from this point on, and correspondingly referred to as the robot's *belief state* at time k .

The probabilistic SLAM approaches, including the EKF, predominantly estimate the belief $b_k(\mathbf{x}_{r_k}, M)$ using some form of *Bayes filter* [35] (which is a temporal extension of the archetypical *Bayes rule* [36]). In doing so, they often treat the SLAM problem as a Markov process through which it is assumed that the current belief state, $b_k(\mathbf{x}_{r_k}, M)$, depends only on the immediately preceding state, $b_{k-1}(\mathbf{x}_{r_{k-1}}, M)$, independent of how the preceding state was reached. The belief probability can therefore be calculated recursively, as shown by the generic Bayes filter:

$$b_k(\mathbf{x}_{r_k}, M) = \eta p(\mathbf{z}_k | \mathbf{x}_{r_k}, M) \cdot \int p(\mathbf{x}_{r_k} | \mathbf{x}_{r_{k-1}}, \mathbf{u}_k) b_{k-1}(\mathbf{x}_{r_{k-1}}, M) d\mathbf{x}_{r_{k-1}} \quad (19)$$

where η is a normalisation constant, $p(\mathbf{z}_k | \mathbf{x}_{r_k}, M)$ is a probabilistic measurement model, and $p(\mathbf{x}_{r_k} | \mathbf{x}_{r_{k-1}}, \mathbf{u}_k)$ is a probabilistic motion model.

However, there are several problems with implementing equation (19) in its generic form. To begin with, the potentially high dimensionality of the map can make the estimation of the belief $b_k(\mathbf{x}_{r_k}, M)$ computationally intractable.

This is difficult to avoid in practice, as the number of landmarks in the map can easily be in the order of hundreds or even thousands. In addition, the belief function is hard to factorise due to the uncertainty in the robot and landmark positions being intricately intertwined [12, 14]. The need to maintain these intricate correlations only complicates the task of addressing the high computational complexity. Another problem to consider is that the full posterior over a continuous space possesses infinitely many dimensions, which cannot be represented by a digital computer [37].

Thus, working instantiations of Bayes filter are the product of additional assumptions and approximations. It is primarily these assumptions, along with their implications, that differentiate the currently existing probabilistic approaches. The type of assumptions adopted shape the strengths and limitations of each approach, which will become apparent in the following reviews.

The *expectation maximisation (EM)* approach, proposed in [38], solves the SLAM problem by estimating the mode of the posterior $p(M | \mathbf{z}_{0:k}, \mathbf{u}_{0:k})$ (also denoted $b_k(M)$ for notational convenience) to find the most likely map M^* , along with the most likely path taken by the robot. Formally, this can be expressed as solving the maximum likelihood (ML) estimation problem

$$M^* = \underset{M}{\operatorname{argmax}} b_k(M). \quad (20)$$

The posterior $b_k(M)$, based on the derivation given in [38], can be written as

$$\begin{aligned} b_k(M) &= \int b_k(\mathbf{x}_{r_k}, M) d\mathbf{x}_{r_k} \\ &\propto \int \cdots \int \prod_{j=0}^k p(\mathbf{z}_j | \mathbf{x}_{r_j}, M) \cdot \\ &\quad \prod_{j=1}^k p(\mathbf{x}_{r_j} | \mathbf{x}_{r_{j-1}}, \mathbf{u}_j) d\mathbf{x}_{r_1} \cdots d\mathbf{x}_{r_k} \quad (21) \end{aligned}$$

where the robot's initial pose is, arbitrarily, set to $\mathbf{x}_{r_0} = [0 \ 0 \ 0]^T$. This equation is void of any constants, normalisation or otherwise, as the objective is to only maximise the posterior $b_k(M)$, not to calculate its true value.

The main problem with solving equation (20) is the high computational complexity. The maximisation of the likelihood function, defined in equation (21), involves searching in the space of all maps, and in each map, integrating over all possible poses at every instant in time. Since this is generally not feasible, an optimisation technique that performs local hill-climbing in likelihood space is used. This technique, based on the classical EM algorithm [39, 40], involves iterating two steps: an *expectation step*, or *E-step*, and a *maximisation step*, or *M-step*.

1) *E-step*: In this phase, probabilistic estimates for the poses $\mathbf{x}_{r_0}, \dots, \mathbf{x}_{r_k}$ are calculated based on the currently best map M and data $D_{0:k}$ (where $D_{0:k} = \{\mathbf{z}_{0:k}, \mathbf{u}_{0:k}\}$).

This can be considered a low-dimensional localisation problem, which is solvable using standard *Markov localisation* [41]. However, there is a slight difference that needs to be considered. Each posterior $p(\mathbf{x}_{r_j} | D_{0:k})$ is estimated using the data from the entire time interval $\{0, \dots, k\}$, which requires two localisation passes: one forwards in time, giving $p(\mathbf{x}_{r_j} | D_{0:j})$, and the other backwards in time, giving $p(\mathbf{x}_{r_j} | D_{j+1:k})$.

2) *M-step*: The maximisation step involves calculating the most likely map M^* based on the pose estimates obtained in the E-step. In essence, this is a map optimisation problem whereby the robot's poses $\mathbf{x}_{r_{0:k}}$ are treated as latent variables. As given in [35], the map M^* is calculated by maximising the expectation over the joint log-likelihood of the robot's path $\mathbf{x}_{r_{0:k}}$ and data $D_{j+1:k}$:

$$M^{[\gamma+1]} = \underset{M}{\operatorname{argmax}} E[\log p(\mathbf{x}_{r_{0:k}}, D_{0:k} | M^{[\gamma]}) | D_{0:k}] \quad (22)$$

Here the superscript ' $[\gamma]$ ' denotes the iteration of the optimisation algorithm. The algorithm generates a sequence of maps $M^{[0]}, M^{[1]}, M^{[2]}, \dots$ with monotonically increasing likelihood until a local maximum is reached. Since finding this local maximum can still be a difficult problem in a high-dimensional space, it has become common practice to represent the map as a discrete *occupancy grid* [42, 43] and solve equation (22) for each grid cell independently.

The EM approach has several key advantages over the EKF. Firstly, it provides a solution to the data association problem that does not require the unique identification of landmarks; in fact observed landmarks can be somewhat indistinguishable. Data association is performed through gradual reinforcement or degradation of matching probabilities as all the observation data over time is considered. This allows past data association decisions to be revised and possibly corrected. The EM approach can also estimate the mode of complex posteriors, and does not assume Gaussian noise like the EKF.

However, there are a few weaknesses that need to be considered. Unlike the EKF, the EM approach processes the entire data set multiple times, and as a consequence, does not provide an incremental solution to SLAM where a map is incrementally built as new observation data is received. Another weakness is that the EM algorithm is traditionally suited to offline processing. Online versions have been proposed (e.g., [44]), however, they partially sacrifice the robustness in the data association process to accommodate the restricted computational time windows. Lastly, the EM approach can become trapped in a local maxima and, hence, arrive at a suboptimal solution.

Montemerlo *et al.* [45] have recently proposed a new probabilistic approach to SLAM that is based on *particle filtering* [46, 47, 48]. This approach, called FastSLAM,

estimates the posterior $p(\mathbf{x}_{r_{0:k}}, M | \mathbf{z}_{0:k}, \mathbf{u}_{0:k})$ (also denoted $b_k(\mathbf{x}_{r_{0:k}}, M)$), which is a slight variation of the commonly sought posterior given in equation (19). That is, instead of estimating the posterior over momentary robot poses, it estimates the posterior over robot paths. Before we describe how this is done, the topic of particle filtering will be briefly summarised first (see [47] for a comprehensive review).

The idea behind particle filtering is to approximate the posterior density in a Markov chain through a process known as *importance sampling* [49]. In essence, the posterior is represented by a set of m random sample states or particles $S_k = \{s_k^j | 1 \leq j \leq m\}$ drawn from it. Each particle is given a weighting ω_k^j called an *importance factor*, which signifies the particle's quality relative to the other particles. The weighted particle set S_k is then processed *in lieu* of the full posterior. Note that the full posterior can still be roughly reconstructed, e.g., using a *histogram* or *kernel based density estimation technique* [50], because of the duality between particles and the posterior from which they are drawn.

A vanilla particle filter process can be described abstractly as follows. First, the initial set of particles S_0 are randomly drawn from the state space. For time $k \geq 1$, the set S_{k-1} is filtered (i.e. transformed into S_k) by computing two stages: a *prediction stage* and an *update stage*. In the prediction stage, a particle \bar{s}_k^j is generated for each particle $s_{k-1}^j \in S_{k-1}$ according to an actuation model. The resulting particles have a distribution commonly referred to as the *proposal distribution*. These particles are placed in a temporary set \bar{S}_k . In the update stage, the weight of all the particles in this temporary set are re-evaluated based on the latest observation information to produce the *target distribution*. Finally, m particles are drawn (with replacement) from \bar{S}_k to give S_k . This involves drawing the higher weighted particles from \bar{S}_k and resampling the others. The specifics of this process, however, vary according to the application and the particular particle filter used.

In the context of mobile robot localisation, the particle filter approach known as *Monte Carlo localisation* (*MCL*) [51] has been shown in studies such as [52] to be more robust than the EKF. MCL can also represent complex posteriors; solve the kidnapped robot problem; and operate as an *anytime algorithm* [53] under limited computational resources. However, for years particle filters were confined to these low-dimensional problems, due to the number of particles needed to populate a d -dimensional space increasing exponentially with d . Particle filters were therefore too inefficient to be used for high-dimensional problems like SLAM. However, this changed when Murphy [54] identified a structural property of SLAM that could be exploited to develop an efficient particle filter. This structural property is based on the condition that correlations in the uncertainty of different map landmarks arise only from uncertainty in the robot's pose. Therefore, if hypothetically the robot knows its trajectory perfectly,

the landmark states can be estimated independently of each other. This conditional independence has led to the use of the so-called *Rao-Blackwellised particle filter* [47] (named after its relation to the *Rao-Blackwell theorem* [55]), which analytically marginalises out some of the variables attributed to a problem's structure for an efficient solution.

The FastSLAM approach is an instantiation of the Rao-Blackwellised particle filter. It uses the structural property identified by Murphy to estimate the posterior $b_k(\mathbf{x}_{r_{0:k}}, M)$ in the factorised form [45, 56]:

$$b_k(\mathbf{x}_{r_{0:k}}, M) = b_k(\mathbf{x}_{r_{0:k}}) \prod_{i=1}^n p(\mathbf{x}_i | \mathbf{x}_{r_{0:k}}, \mathbf{z}_{0:k}) \quad (23)$$

This factorisation is exact and universal to the SLAM problem. Essentially, it decomposes the posterior over robot paths and maps into $n + 1$ recursive estimators: one estimator over robot paths $b_k(\mathbf{x}_{r_{0:k}})$ and n separate estimators over landmark states $p(\mathbf{x}_i | \mathbf{x}_{r_{0:k}}, \mathbf{z}_{0:k})$ conditioned on each hypothetical path.

FastSLAM estimates the robot path posterior $b_k(\mathbf{x}_{r_{0:k}})$ using a particle filter. Each of the particles in this filter maintains its own map that consists of n independent EKFs, one for each of the landmarks. Thus, the j -th particle at time k can be written in the form

$$S_k^j = \{ \mathbf{x}_{r_k}^j, \underbrace{\mu_{1,k}^j, \Sigma_{1,k}^j}_{\text{landmark } \mathbf{x}_1}, \dots, \underbrace{\mu_{n,k}^j, \Sigma_{n,k}^j}_{\text{landmark } \mathbf{x}_n} \} \quad (24)$$

where the mean $\mu_{i,k}^j$ and covariance $\Sigma_{i,k}^j$ are the Gaussian parameters of each landmark posterior. Therefore FastSLAM integrates particle filtering with Kalman filtering; however, in this context, each EKF is only estimating a single landmark position and, hence, is low-dimensional.

Currently, there are several variants of the FastSLAM algorithm, including *FastSLAM version 1.0* [45], *2.0* [57], and FastSLAM with unknown data association [56]. Assuming that the data association is uniquely known and the initial set S_0 has been initialised, the filtering algorithm proceeds as follows. First, the path posterior is extended by sampling a new pose \mathbf{x}_{r_k} for each particle in the prior sample set S_{k-1} . FastSLAM 1.0 samples new poses based on the most recent control input \mathbf{u}_k :

$$\mathbf{x}_{r_k}^j \sim p(\mathbf{x}_{r_k} | \mathbf{x}_{r_{k-1}}^j, \mathbf{u}_k) \quad (25)$$

Although the measurements \mathbf{z}_k are ignored, they are later incorporated through the resampling process. Nevertheless, this way of sampling new poses has been identified as being inefficient [57], especially when the robot's motion errors are large relative to the measurement noise. When this is the case, sampled poses tend to fall into areas of low measurement likelihood and are consequently poorly weighted. It is then likely that a large proportion of the sampled poses will be terminated, or wasted, through the resampling process.

FastSLAM 2.0 addresses this problem by incorporating the measurements into the proposal distribution

$$\mathbf{x}_{r_k}^j \sim p(\mathbf{x}_{r_k} | \mathbf{x}_{r_{0:k-1}}^j, \mathbf{z}_{0:k-1}, \mathbf{u}_{0:k-1}) \quad (26)$$

which constitutes the primary difference between the two versions.

The second step of the filtering process involves updating the observed landmark estimates. This is performed by linearising the measurement function \mathbf{h} and applying the standard EKF measurement update equations [13] (refer to [56] for a detailed description). These first two steps are then repeated m times to produce a set of m particles. The final step involves correcting the proposal distribution through resampling. Each particle is first assigned an importance weight, given by

$$\omega_k^j = \frac{\text{target distribution}}{\text{proposal distribution}} \quad (27)$$

Then m particles are drawn (with replacement) with a probability proportional to their weights. In the case of FastSLAM 1.0, this resampling process accounts for the latest measurements \mathbf{z}_k , which were earlier ignored. The purpose of resampling in FastSLAM 2.0, however, is more mundane. It is used merely to correct mismatches in the normalisation between particles [57].

In terms of performance, the FastSLAM approach has several key strengths. First and foremost, data association decisions can be robustly made on a per-particle basis, analogous to multiple hypothesis tracking (MHT) (discussed in section II-A). Therefore, instead of just maintaining the data association with the maximum likelihood, the posterior tracks multiple data associations that are resolved over time. Another strength is its computational complexity of $\mathcal{O}(m \log n)$ when the maps are represented by binary trees [45], which is theoretically lower than the quadratic complexity of the vanilla EKF. Also, FastSLAM can cope with a non-linear vehicle model without the need for linearisation, and it can solve the kidnapped robot problem.

The primary weakness of FastSLAM is that the resampling process continually reduces the diversity in the particle set by repeatedly discarding some particles and duplicating others [58]. If the resampling steps of every particle is traced back in time, there will be a point at which all the particles share a common history of the robot's trajectory and hence the same ancestor. Therefore the hypotheses of the robot's trajectory and landmark positions prior to this point of commonality cannot be revised. The resulting lack of particle diversity, called the *impoverishment problem* [59], restricts the size of the *loop* or exploratory excursion that can be corrected (the concept of *closing the loop* will be fully described in section III) and can lead to a suboptimal solution. A closely related problem occurs when there is an insufficient number of particles in the vicinity of the correct state [60]. This deprivation problem, which is inherent to all proactive approaches, is especially troublesome in large environments or when the robot is proverbially kidnapped. A lack of particles or the

time lag involved in distributing particles can ultimately cause the filter to diverge. While increasing the number of particles can offset these problems, it adds to the computational complexity and therefore there is a limitation to the amount of particles that can be processed in real-time. Finally, a large computational effort can be wasted in updating particles with a negligible weight, as the variance of the weights tends to increase stochastically over time.

C. Scan Matching

Another category of SLAM approaches includes those that are based on aligning neighbouring sensor scans, e.g., from a laser or sonar scanner, to estimate the relative translations and rotations of the robot between scans. These *scan matching* approaches align the overlapping segments of the scan set by minimising some distance metric between inter-scan primitives or raw data. This is somewhat similar to *model-based matching* [61], however, scan matching does not use an accurate, dependable model as the base for comparison. Instead, it finds the congruence between noisy data sets that are negatively affected by occlusion and hence the robot's limited field of view.

The majority of scan matching approaches are derived from the *Iterative Closest Point (ICP) algorithm* [62, 63] and its many variants [64]. These approaches are based on iteratively refining an initial robot pose estimate obtained through odometry, which limits the search space. However, it is assumed that the displacement between the initial estimate and the robot's true pose is small enough to arrive at the globally optimal match.

The various approaches mainly differ in the primitives they select and match; the type of distance metric used (e.g., sum of squared distances between corresponding pairs); the weighting of correspondences; and the rejection of outliers. For example, Cox matches scan points to the line segments of a hand-crafted map [65]. Lu and Milios matches points to points in an *a priori* unknown, arbitrary environment (not necessarily polygonal) [66]. Their method does not rely on the uniqueness of landmarks and derives robustness from using the bulk of the scan points in the matching process. Gutmann and Konolige use a combination of the above two methods to take advantage of the computational efficiency of Cox's method and the universal capabilities of Lu and Milios's method [67]. They also take into consideration the topological relationships between neighbouring robot poses, associated via odometry and scan overlaps, to maintain a consistent map in large cyclic environments. Jensen and Siegwart establishes correspondences between points based on a probabilistic distance metric that incorporates both sensor noise and robot pose uncertainty [68]. This provides a way of robustly detecting outliers, and as a result, their algorithm exhibits a faster convergence than the standard ICP algorithm.

There are other, less common types of scan matching approaches. Some approaches are based on finding statistical correlations between scans, such as Weiss and Puttkamer's histogram matching approach [69] and Biber's *normal*

distributions transform (NDT) [70]. These approaches do not require explicit correspondences between individual scan elements, however, they rely on the chosen statistical criteria effectively modeling the environment.

There are also approaches that have the ability to globally localise the robot without the aid of initial pose information. Crowley *et al.* accomplish this by using a training set of range scan profiles from various known poses to construct a lookup table, which can then be indexed to identify possible origins of a scan [71]. Gutmann *et al.* exploit the structured nature of the *RoboCup soccer field* [72] to match line segments extracted from a scan to those of an *a priori* map [73]. Weber *et al.* similarly use an *a priori* map of a structured environment, but instead of matching line segments, they match edges and concave/convex corners [74]. The matching process involves heuristically searching for corresponding patterns of inter-feature relationships, which are invariant to the robot's observational viewpoint. Tomono matches what are called *directed points*, comprising points and their tangent directions, which are also viewpoint invariant [75]. His approach solves the SLAM problem; however, it is computationally complex and the map does not converge over time. To address the complexity problem, global localisation is only applied when the robot fails to find a match using a localised search in the vicinity of the odometry estimate. This can lead to a suboptimal solution because large odometry errors in a partially symmetrical environment can produce multiple hypotheses which all need to be considered. Nevertheless, multiple hypotheses can only be resolved if the environment has unique, detectable features and, hence, is not overly symmetrical.

Finally, there are hybrid approaches that combine some other SLAM approach with scan matching. For instance, Hähnel *et al.* combine FastSLAM with scan matching to minimise odometry error, thereby reducing the number of particles needed to build large-scale maps [76]. Pradalier and Sekhavat [77], on the other hand, use scan matching to improve the data association robustness of an EKF variant called the *geometric projection filter (GPF)* [78].

D. Qualitative Approaches

The last category of SLAM approaches in many ways mimics the qualitative, relativistic knowledge used in an animal or human's mental representation (or *cognitive map* [79]) of navigable environments [80, 81, 82, 83]; and hence has a biological premise. Qualitative SLAM approaches obviate the need for rigorous models of the robot's locomotion mechanism and sensors. They also do not strive for a metrically accurate map, which in combination gives them a heightened robustness and computational efficiency. These approaches, including [84, 85, 86], observe the topological spatial relationships between landmarks or obstacles to navigate and map the environment. Since their main objectives does not include finding absolute position estimates, they are in a sense the least relevant in this review to the new SLAM approach proposed in the next section; and for this reason, they will not be reviewed in any

depth here. However, qualitative approaches do highlight the inverse relationship between rigorous modeling and robustness/generalizability, which is an important justification for the proposed approach.

III. THE KIDNAPPED WAY

This section provides a comprehensive blueprint of the proposed SLAM approach. The underlying principles and ideas of this approach are described, along with the various intricacies of solving the SLAM problem without odometry or the assumption of motion continuity. To begin, there is a brief discussion in Subsection III-A of the requirements and makeup of the sensor system and landmark detection process. This is followed by the multiple-hypothesis data association algorithm in Subsection III-B. Subsection III-C describes the representation and handling of positional errors for constructing the environmental map. An interesting temporal anomaly is also considered, along with well-known SLAM issues such as the *correlation problem*, *map convergence*, *map consistency* and *closing the loop*. Subsection III-D explores a few pathological or failure cases, and lastly Subsection III-E describes the mechanisms for incorporating dead-reckoning information and the resulting performance gains. Note that there are no results from practical experimental trials given here, however, preliminary results are given in [87, 88] and more conclusive results gathered from several national parks and bushlands² will be published shortly.

Before we begin, the pool of available English/Greek letters for use as mathematical symbols has been exhausted in the previous section and therefore several letters will have to be reassigned. This will be done in an explicit manner, when needed, to avoid any confusion.

A. Landmark Detection

The landmark detection process involves using a sensor system to perform scans of the environment, and in each scan, observe a batch of landmarks and their relative geometric relationships. Since these relationships tend to remain invariant to the observer's point of view [89], they can be exploited to recognise landmarks whose appearance would otherwise vary according to the robot's changing pose. The sensor system used in this role can comprise one of many possible *active* or *passive* sensing devices [90, 91, 92] or a fusion of multiple devices in a complementary or redundant fashion [26, 93]. Each option has comparative advantages and disadvantages. However, because exteroceptive based sensing is such a critical element of this SLAM approach, there are several sensor qualities that are either necessary or, at the very least, highly desirable.

Firstly, the sensor needs to be able to gather a rich, dense data set in each scan that adequately represents a collection of landmarks. The more information that can be gathered to distinguish the individual landmarks, e.g., size, shape, colour, inter-landmark distances and angles, the less

combinational matches that exist between the landmarks in a scan and their possible counterparts in the map and, hence, the more efficient the data association process. Secondly, during the scanning interval, the robot is assumed to be either stationary or have a negligible movement with respect to the scanning speed. This is a requirement because without dead-reckoning information, motion compensation cannot be applied to the data set. Also, in the case of 2D SLAM, the sensor may need to either remain in an upright position, e.g., by being suspended on a gimbal, or ascertain its roll and pitch angles through auxiliary sensors, such as an inclinometer or gyroscope, to track the horizontal plane when the robot traverses over an undulating surface.

An important objective to consider is that the angle of view and range / depth of field of the sensor need to be maximised in order to ensure a high probability of overlap between the scan and map, and to maximise the overlapping region itself. There are also domain considerations that dictate the suitability and capacity of different sensing modalities. In a natural outdoor environment, for instance, a laser scanner can be used to accurately determine the surrounding geometry, whereas a digital colour camera may not be able to reliably differentiate between the many subtle shades of colour due to changing lighting conditions.

To provide a representative example, the authors implemented the proposed SLAM approach for outdoor environments using the 3D laser scanner proposed in [94]. This scanner consists of a laser rangefinder (LaserAce IM HR from MDL, UK) mounted on a pan-tilt unit (PTU-46-17.5 from Directed Perception, CA, USA) for 3D angular positioning. The laser rangefinder has maximum range of 300m; a resolution of 1dm; a typical accuracy of 3dm; and makes range measurements at a rate of 1000Hz. A 3D scan is performed by using the pan-tilt unit to horizontally slue the laser in a back-and-forth manner, while incrementally adjusting its elevation angle. The scan specifications are based on the selection of values for the horizontal angular range, horizontal slue rate, vertical angular range and vertical resolution parameters of the pan-tilt unit. For example, scans in a recent experimental trial were performed using a horizontal range of 257°, horizontal slue rate of 51.4°/s, vertical angular range of 25.7° and vertical resolution of 1.03°, which resulted in each scan taking 130,000 readings over a time period of approximately 2.2 minutes. Note that this number of readings per scan can be dramatically increased to millions (e.g., by decreasing the horizontal slue rate and increasing the vertical angular resolution); however, this is at the cost of a higher acquisition time and memory/computational load.

After each scan, the landmark detection algorithm extracts a batch of landmarks from the collected data set. The algorithm proposed here is fully described in [87]. It extracts a set of 2D point landmarks, in the sensor's local frame of reference (see Fig. 1), that represents the centroids of environmental features whose spatial extent is orthogonal to the sensor's horizontal plane. This involves searching for arbitrary features (of no particular shape) that

²A term used to describe the harsh Australian outback.

are highly visible, laterally compact, and not partially occluded by other features. The first two criteria are based on relative measures. That is, highly visible features are those that occupy a large spatial range along the z -axis relative to their neighbours. Lateral compactness is a measure of how small a feature extends in the horizontal direction. Features that are more laterally compact than others tend to have a lower variability in their perceived centroid when viewed from different angles, and therefore provide more accurate triangulation results. The third criterion, however, removes unstructured features that are partially hidden behind other features, as their true appearance cannot be ascertained.

The features that meet these criteria in an outdoor environment may include, amongst others, trees, bushes, plants, poles/posts, buildings, rock formations and sharp surface variations. A *convex hull* [95, 96] is created around each of these features to extract their centroid points, which provides the landmark set $S_k = \{s_{j,k} \mid 1 \leq j \leq m_k\}$ at time k . The number of landmarks detected m_k must be at least three to obtain a unique triangulation result in 2D SLAM. However, preferably more than this critical number is detected for the purpose of redundancy.

There are several issues about this type of landmark set that need to be considered. Firstly, the landmarks are only represented by 2D points in local coordinates

$$s_{j,k} = \begin{bmatrix} x'_{j,k} \\ y'_{j,k} \end{bmatrix} \quad (28)$$

and therefore there is no other information used, such as landmark size, shape, texture or model based identification, to assist the data association process. It should be noted, however, that this landmark detection algorithm serves as a generalised basis for the proposed SLAM approach, to which there are many extensions (as discussed in Section IV). Secondly, by using a convex hull to determine a landmark's centroid, the resulting centroid point may not lie within the landmark itself if the landmark is concavely shaped. While having a centroid point in free space is theoretically incorrect, it does not affect the workability of this representation. Thirdly, there is a phenomenon that we call "the wandering centroid" which occurs as a result of a solid landmark being partially hidden from the sensor's field of view. As the robot moves around the landmark, as depicted in Fig. 2, the perceived centroid tends to wander with it. The extent of this deviation becomes increasingly more pronounced as the robot moves towards the landmark or as the landmark's lateral size increases, thereby exacerbating the occlusion. This error, along with sensor noise, can be taken into account using a probabilistic model or some type of gating technique. In this case, a simple Euclidean error distance ϵ is used as a tolerance bound for each centroid.

A landmark that can be hidden to a considerable degree from the sensor's field of view may be represented by multiple centroids over time, as shown in Fig. 3 (note that the landmark shown here is structured only for illustrative purposes). In essence, each of these centroids represents

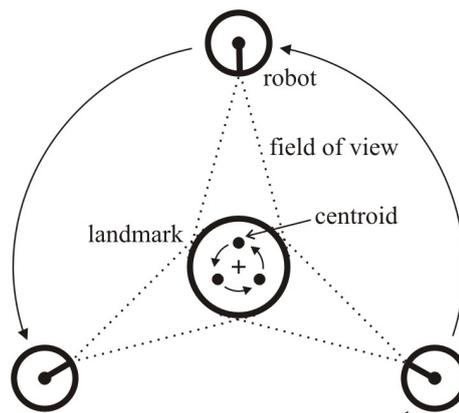


Fig. 2
THE WANDERING CENTROID

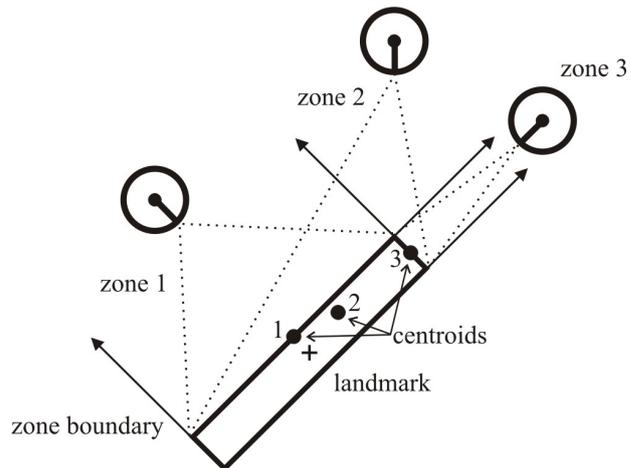


Fig. 3
PARTITIONING A LANDMARK ACCORDING TO VIEWING AREAS

an individual landmark whose appearance is restricted to when the robot is in a particular viewing area or zone. The number of centroids can be reduced and the zones expanded, by increasing the error tolerance ϵ . However, this increases the possible data association combinations in a cluttered environment; and unless there is a dominance of laterally large or elongated features, it is highly unlikely that any single feature will generate a dense cluster of insignificant landmark points anyway.

B. Multiple-Hypothesis Data Association

The multiple-hypothesis data association algorithm matches the batch of landmarks extracted from the scan S_k to those in the map M , while considering any ambiguities that arise from environmental symmetries or sensor limitations. Since at time $k = 0$ the environment is unknown, the first scan S_0 can be used to provide both the global

frame of reference, as shown in Fig. 1, and the initial set of landmarks in M . The robot's pose is also initialised, somewhat arbitrarily, to $\mathbf{x}_{r_0} = [0 \ 0 \ 0]^T$. The proposed data association algorithm is then executed for time $k \geq 1$. We shall first describe this algorithm based on the scan $S = \{s_1, \dots, s_m\}$ (the time subscripts have been left out for brevity) and map $M = \{x_1, \dots, x_n\}$ at time $k = 1$, and then show how *multiple hypothesis tracking (MHT)* is incorporated for future times.

Unlike most other SLAM approaches, data association in this approach cannot be performed using the standard *gated nearest-neighbour (NN) algorithm* [2], because without dead-reckoning or motion continuity information, the required pose predictions cannot be made. Instead, we propose a graph matching approach that transforms the scan S and map M into two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, respectively, and then finds subgraph isomorphisms between them to determine their commonalities. The vertex sets V_1 and V_2 of the graphs are used to represent the individual landmarks, which in this case are only distinguished by 2D points. The edges that interconnect the vertices, given by the sets $E_1 \subseteq V_1^2$ and $E_2 \subseteq V_2^2$, are used to represent geometric relationships that are invariant to the robot's viewpoint. In doing so, common subgraphs of the graphs G_1 and G_2 , and hence data association hypotheses, can be found independently of the robot's pose. The geometric relationships that are used here are the Euclidean distances between landmark points. While these relationships may not be as discriminant as others, e.g., the relative landmark orientations [32], their detection does not rely on structured features being in the environment.

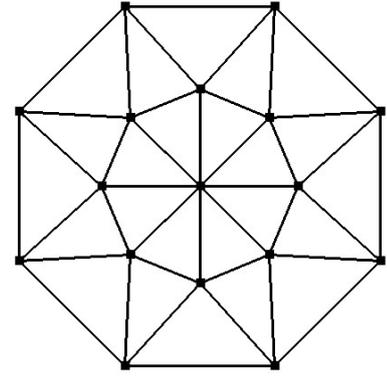
A data association hypothesis can now be given by the correspondence set $X \subseteq V_1 \times V_2$, where the vertices in each of the matched pairs $\langle s_a, x_b \rangle \in X$ share a consistent set of edges. Since the graphs are matched based on their edges, there is a complexity problem that arises if the graphs are complete (i.e., fully connected). While the scan graph G_1 remains relatively small, the map graph G_2 is dynamically expanding over time, causing an exponential growth in the number of edges.

For a complete map graph with n vertices, the number of edges is given by the binomial coefficient

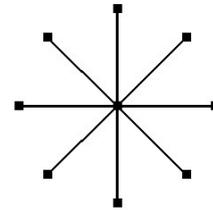
$$\binom{n}{2} = \frac{n(n-1)}{2}. \quad (29)$$

Therefore, the number of edges in a graph with 10 vertices is 45; 100 vertices is 4950; or 1000 vertices is 499,500. To reduce this growth rate, a nearest-neighbour strategy is proposed that confines the inter-landmark relationships to only those between neighbouring landmarks. This strategy is based on the *Delaunay triangulation (DT)* [97], which can optimally generate a triangular mesh from a point set in $\mathcal{O}(n \log n)$ time. An example of a DT is shown in Fig. 4(a).

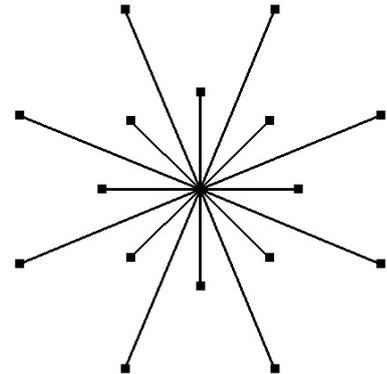
First, a DT is created for both the landmark sets S and M . The resulting mesh edges, indicating the closest landmark neighbours, form the initial edge set of the



(a) Delaunay Triangulation (DT)



(b) 1st Neighbour Depth of the Center Vertex



(c) 2nd Neighbour Depth of the Center Vertex

Fig. 4

NEAREST-NEIGHBOUR STRATEGY

corresponding graphs G_1 and G_2 , respectively. The vertices of these graphs are now linked to their most immediate neighbours, which we call a depth level of one (see Fig. 4(b)). To obtain a depth of two, edges are added to the graphs that link landmarks that are separated by a path length of two in the DTs (see Fig. 4(c)). Similarly, a depth of three is obtained using a path length of three; and so on. The selection of which depth level to use is therefore a tradeoff between graph completeness and computational complexity. The depth level can also be changed over time, e.g., decreased to compensate for an expanding map, or different values can be used for each of the vertices, as a measure of their importance or uniqueness, to regulate their contribution to the matching process.

To demonstrate how the number of edges varies according to the chosen depth, a simulation was performed that

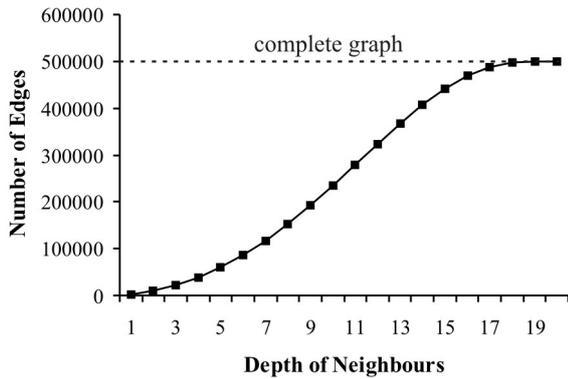


Fig. 5

THE EDGE COUNT AS A FUNCTION OF NEIGHBOUR DEPTH FOR A 1000-VERTEX GRAPH

generated 1000 random points on a 2D plane. The nearest-neighbour strategy was then applied, with the depths ranging from one to twenty for all vertices. The resulting edge counts are graphed in Fig. 5. As an example, a depth of two resulted in 9784 edges, which is roughly 2% of the number in a complete graph.

Apart from a selectable reduction in the number of edges, an important property of this strategy is that it can adapt to changes in the density and separation of landmarks. Therefore, a landmark's nearest neighbours is determined by the spatial attributes of its local cluster. Note that its nearest neighbours is also influenced by the DTs optimality criterion, i.e., maximising the minimum angle of the triangles; however, we conjecture that other triangulation schemes can be used with similar results.

A depth of two will be used here for both the scan graph G_1 and map graph G_2 . Hence, the only vertices of these graphs that will be interconnected by edges, representing Euclidean distances, are those that satisfy the nearest-neighbour strategy at a depth of two. An example of the resultant graphs G_1 and G_2 is shown in Figs. 6(a) and 6(b), respectively. (Note that the edge distances are to scale.) While these graphs happen to be complete in this instance, due to their small size, their primary function is to provide a simple example for describing the matching process.

The first step of the proposed matching algorithm involves creating a *correspondence graph* [98], denoted C , which represents the compatibility between each of the pairs $\langle s_a, x_b \rangle \in V_1 \times V_2$. The vertices of graph C are all the pairs whose two elements share a common property, i.e., a connecting edge of equal distance (within tolerance bounds). The edges of C represent the consistency between each of these vertices. That is, if $\langle s_1, x_2 \rangle$ and $\langle s_3, x_4 \rangle$ are two vertices of C , they are interconnected by an edge if the distances associated with $E_1(s_1, s_3)$ and $E_2(x_2, x_4)$ are equivalent. The method for creating graph C involves first finding all the edge matches between graphs G_1 and G_2 , and then adding the possible vertex combinations of

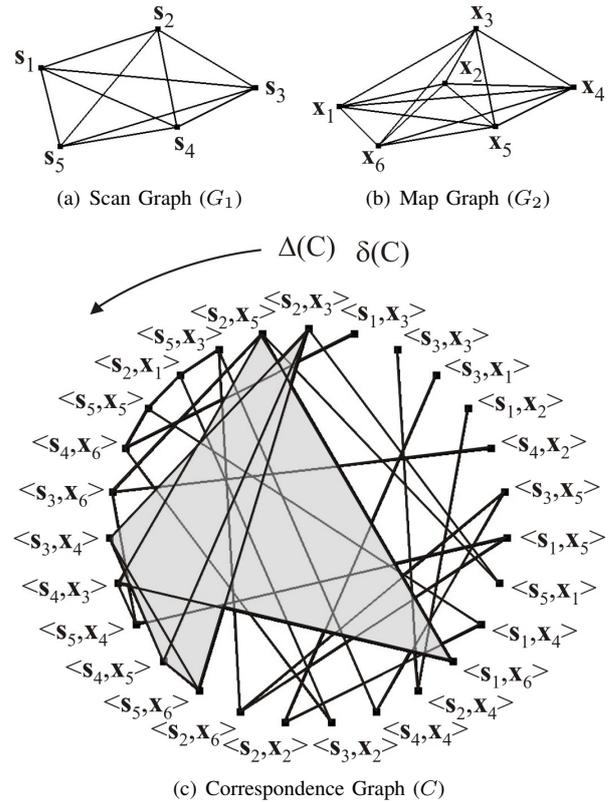


Fig. 6

COMMON SUBGRAPHS OF THE GRAPHS G_1 AND G_2

each edge match to C as vertices and interconnecting those that are compatible with edges. As an example, the edge $E_1(s_2, s_3)$ of G_1 in Fig. 6(a) is a match for the edge $E_2(x_3, x_4)$ of G_2 in Fig. 6(b). The possible vertex combinations, and hence the resulting vertices in C , include $\langle s_2, x_3 \rangle$, $\langle s_2, x_4 \rangle$, $\langle s_3, x_3 \rangle$ and $\langle s_3, x_4 \rangle$. The compatible edges are $\{\langle s_2, x_3 \rangle, \langle s_3, x_4 \rangle\}$ and, through symmetry, $\{\langle s_2, x_4 \rangle, \langle s_3, x_3 \rangle\}$. The final state of the correspondence graph after all the matched edges between G_1 and G_2 have been incorporated is shown in Fig. 6(c).

It is common practice to then find the *maximum clique* (maximum complete subgraph) in graph C to obtain the maximum common subgraph (MCS) of the graphs G_1 and G_2 [32]. However, there are several problems with this approach. Firstly, finding the maximum clique of an arbitrary graph is *NP*-complete, and hence is computationally complex [99]. Furthermore, the graphs G_1 and G_2 would have to be complete, which as discussed previously, exacerbates the complexity problem as the map M expands over time. Another problem to consider is that the MCS, while the best match, may not be the right match. The MCS is merely the best hypothesis at one particular time instant; however, another common subgraph may prove to be a better hypothesis over time. Consequently, a new algorithm is proposed here that finds common subgraphs based on the notion that only three consistent vertices in graph C

are required to triangulate the robot's pose.

First, the vertices of graph C are ordered according to their degree (valence) sequence, i.e., in monotonically nonincreasing degrees from the maximum degree ΔC to the minimum δC (see Fig. 6(c)). The reason for doing this is that it heuristically places the vertices that are most likely to belong to a large common subgraph early in the order. Next, the vertices are processed in turn, by following each of their edges in search for a triangular subgraph. To show some examples, several triangular subgraphs are highlighted in Fig. 6(c), such as the subgraph $\{\langle s_2, x_3 \rangle, \langle s_3, x_4 \rangle, \langle s_4, x_5 \rangle\}$.

When a triangular subgraph is found, the *circle intersection approach* [87] is used to triangulate the robot's pose x_{r_k} based on the three constituent vertices. A unique pose is found if the three vertices do not lie on a circle or in a straight line (a conjugate pair is obtained in the case of a straight line). The robot's pose is then used to transform the local coordinates of the scan S into global coordinates:

$$x_j = x_{r_k} + x'_j \cos(\theta_{r_k}) - y'_j \sin(\theta_{r_k}) \quad (30)$$

$$y_j = y_{r_k} + x'_j \sin(\theta_{r_k}) + y'_j \cos(\theta_{r_k}) \quad (31)$$

With both the scan S and map M now being in the same coordinate system, the points in S are directly compared to those in M to obtain the complete set of vertex matches and an associated match count (i.e., the number of vertex matches). This constitutes one hypothesis. The process now reiterates for the next triangular subgraph that is found; however, if it is a subset of the earlier hypothesis, then it can be eliminated, especially if it is associated with higher positional errors (described in the next subsection).

There are two ways in which this algorithm can terminate. The first way is if every triangular subgraph has been found and processed in a brute force manner. The second is as an *anytime algorithm* [53], which takes advantage of the vertex order to arrive at the best hypotheses within the available time window. After completion, the resultant hypothesis set is given by $H_k = \{h_k^{(\lambda)} \mid 1 \leq \lambda \leq \Lambda\}$, where Λ is the number of hypotheses in the set. Note that the superscript ' (λ) ' will also be used for other variables to indicate the particular hypothesis to which they belong.

The hypotheses H_k are handled by the MHT algorithm, which resolves their associated data association ambiguities by tracking and assessing them over time. The MHT algorithm proposed here does not compare hypotheses based on a probabilistic model like in [34]; instead, hypotheses are weighted according to how many data association matches are made over time. Therefore, stronger hypotheses are those whose evolution of the map better corresponds with the accumulated observation data.

Each hypothesis $h_k^{(\lambda)}$ maintains its own version of the map $M^{(\lambda)}$, robot path $x_{r_0:k}^{(\lambda)}$, and vertex match count $w_k^{(\lambda)}$. The hypotheses are compared based on their accumulated

match count $W_k^{(\lambda)}$ over k time periods, given by

$$W_k^{(\lambda)} = \sum_{t=0}^k w_t^{(\lambda)} \quad (32)$$

Initially, one hypothesis $h_0^{(1)}$ is created at time $k = 0$ to hold the map $M^{(1)}$ (from scan S_0), robot pose $x_{r_0}^{(1)}$, and match count $w_0^{(1)}$ (initialised to zero). For time $k \geq 1$, the data association algorithm is executed for each hypothesis in H_{k-1} , which produces a new set of hypotheses H_k . These hypotheses are then compared based on their accumulated match count $W_k^{(\lambda)}$. The best hypothesis, and hence the SLAM solution, is the one with the highest weighting; however, the best Λ_{max} are kept. The map building algorithm is then called for each hypothesis in turn (described in the next subsection). This process then reiterates for the next time instant.

The maximum set size Λ_{max} is used to limit the number of hypotheses being spawned during each cycle, and therefore maintains tractability. Weak hypotheses are consequently discarded in a "survival of the fittest" manner, leaving strong hypotheses to reign. The potential problem with this, however, is that there may be a point in the past where all hypotheses share the same robot path, which as a result, cannot be revised by this algorithm. This is in fact a similar problem to that found when limiting the number of particles in FastSLAM [58]. The result is that the selection of Λ_{max} is a tradeoff between the hypothesis diversity and computational complexity.

To illustrate how this MHT algorithm operates, an example scenario is given in Fig. 7. The initial map $M^{(1)}$ at time $k = 0$ (from scan S_0) is shown in Fig. 7(a). This map has two symmetrical landmark clusters: $\{x_2, x_3, x_4\}$ and $\{x_7, x_5, x_6\}$. A scan at time $k = 1$ is then performed, which provides the landmark set S_1 shown in Fig. 7(b). The three landmarks s_1, s_2 and s_3 have the same spatial properties as either of the symmetrical clusters in map $M^{(1)}$. The landmark s_4 , however, does not correlate with any landmark in $M^{(1)}$, possibly because it was previously obstructed or out of range; sensor errors; or environmental dynamism. In any case, the result is two hypotheses $h_1^{(1)}$ and $h_1^{(2)}$, shown in Figs. 7(c) and 7(d), respectively, which account for two possible positions of the robot and the new landmark x_8 . Since both hypotheses have three landmark matches, they have an equal weighting of three. At time $k = 2$, a new landmark set S_2 , shown in Fig. 7(e), is obtained from a scan. This results in several hypotheses being generated, including $h_2^{(1)}, h_2^{(2)}$ and $h_2^{(3)}$ shown in Figs. 7(f), 7(g) and 7(h), respectively. The best hypothesis at this time is $h_2^{(1)}$, as s_1 reinforces the new landmark x_8 in $h_1^{(1)}$ and s_4 matches the unique landmark x_1 .

This reveals the underlying premise upon which this algorithm is based. That is, data association ambiguities can be resolved if new landmarks can be added to the hypothesised maps now and then observed in combination with other distinguishing landmarks at a later time. There are, however, several caveats. Firstly, if all the unmatched

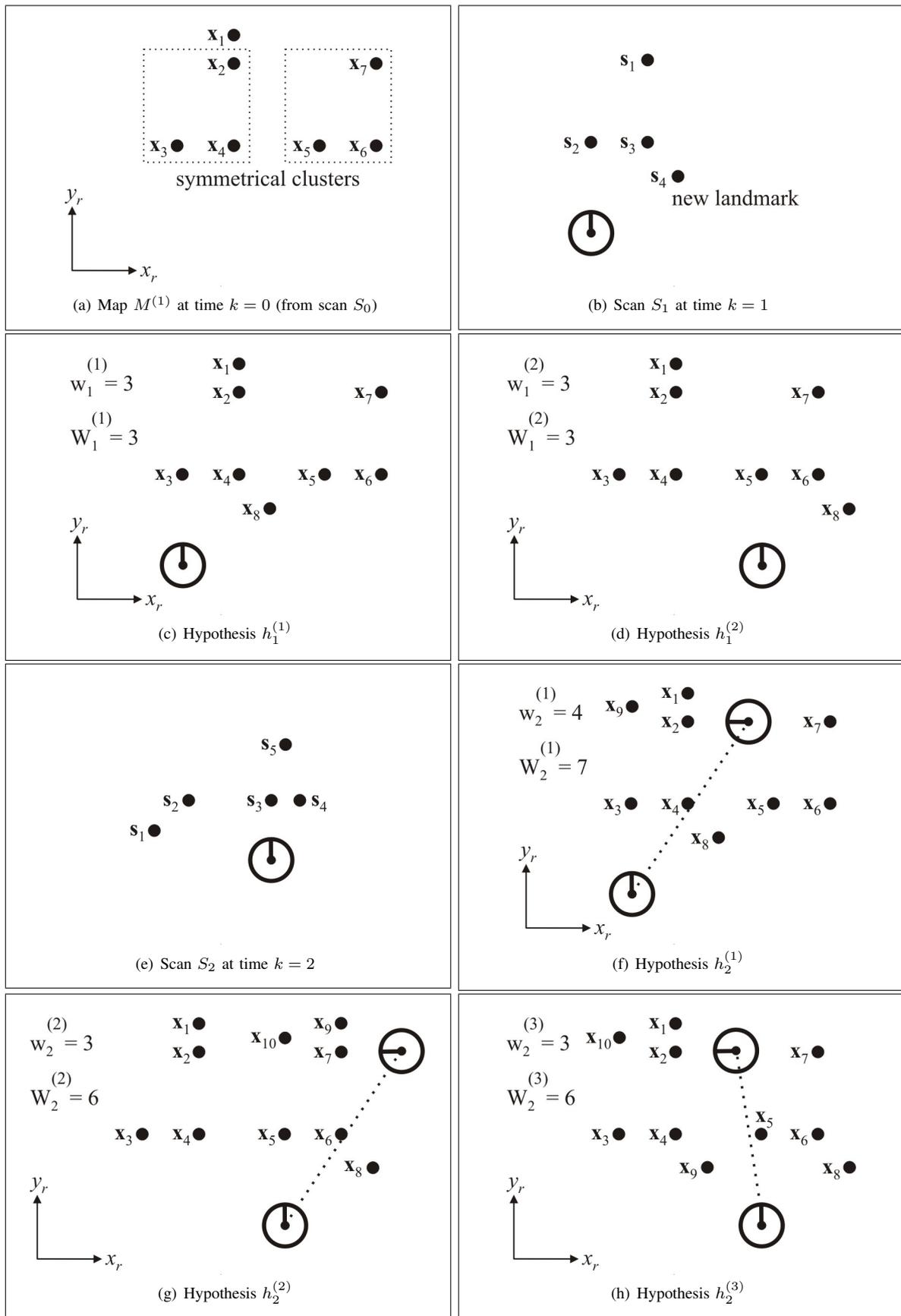


Fig. 7

A MULTIPLE HYPOTHESIS TRACKING (MHT) SCENARIO

landmarks are added to the maps, then weaker hypotheses, by definition, will have larger maps and hence more clutter. This means they will be prone to more false data associations, which can potentially boost their weights. Another caveat to consider is that adding new landmarks across the maps of different hypotheses can create symmetries or mirrors in the landmark patterns, thereby decreasing their differentiability.

However, these issues can be addressed by adding and removing landmarks in a strategic manner. In the case of adding new landmarks, a regulative upper limit on the number added to each map can be used to minimise the variance of the map sizes. The new landmarks can then be carefully selected based on their comparative uniqueness or resolvability. In the case of removing landmarks, the match count of each landmark can be compared to that of its neighbours as an indication of its relative importance. Landmarks of low importance, possibly due to bad placement, limited visibility, movement or being erroneous, can consequently be discarded to reduce map clutter. Also, landmarks can be removed based on what is called *negative information*, which can be gathered from inconsistencies between the landmarks detected in a scan and the expectation of what should have been detected, given the resultant hypotheses.

C. Positional Error Representation and Handling

This subsection presents a novel approach to the representation and handling of positional errors that is considerably different to any of those reviewed in Section II. For instance, there is no use of Gaussian probabilities like in the EKF [12], nor does the multiplicity of samples in one of the Monte Carlo approaches [47] have any discernible similarities. The approach proposed here is based on the sole premise that the positional uncertainty is directly proportional to the radial distance from the origin. That is, the further away a landmark is from the origin, the larger its positional error tends to be.

The idea behind this approach stems from a simplification of the *correlation problem* [12, 14], which is generally the problem of maintaining the robot-landmark correlations that are formed when the robot's imprecisely known pose is used to update the landmark positions and *vice versa*. However, since this SLAM approach is based on perpetually solving the kidnapped robot problem, the uncertainty in the robot's pose at time k is entirely attributable to the landmarks used in the triangulation. Therefore, the only correlations that need to be maintained are between the landmarks themselves.

Each landmark is correlated to the specific landmarks used to triangulate its global position. Consequently, a landmark's positional accuracy is dictated by the accuracy of its correlated landmarks, along with the sensor system and landmark detection process (note that odometry does not play a role here). Now, if we forgo any rigorous error models of the sensor system, and the associated contrivances, then the initial landmarks in the map $M^{(1)}$ (from scan S_0) at time $k = 0$ can be considered to be

the most accurate of all landmarks, as they define the global origin and are not based on any triangulations. Landmarks that are added to the map from scan S_1 at time $k = 1$ are triangulated using three of the initial landmarks, and therefore are based on one triangulation. As a result, their positional uncertainty is higher than the initial landmarks because the scanning process introduces new errors, e.g., sensor noise. Similarly, if the landmarks from scan S_2 at time $k = 2$ are triangulated using three of the landmarks added to the map at time $k = 1$, then they will be based on two triangulations, as their positions are derived from a triangulation of a triangulation. Again, these landmarks have a higher positional uncertainty than their correlated landmarks because the scanning process introduces additional errors. From this, we can conclude that the number of triangulations from which a landmark's position is based is associated with an accumulative error. Therefore, given that the sensor system has a limited range, landmarks further away from the origin will tend to be based on more triangulations and, consequently, have larger positional errors; hence our initial premise.

The way in which positional errors are handled here is analogous to solving the well-known *traveling salesman problem (TSP)* [100, 101], which is the problem of finding the shortest closed path between n cities, given their intermediate distances. In this case, however, the problem is defined as finding the shortest path between each of the n landmarks and the origin, where the path length is given by the number of triangulations from which a landmark's position is derived. Therefore the objective is to minimise the number of triangulations used to reach every landmark, and in doing so, minimise the sequential transfer and distortion of the primary information (i.e., the initial landmarks).

The error in each of the landmark positions $\mathbf{x}_i^{(\lambda)}$ is represented by a single nonnegative integer $\xi_i^{(\lambda)} \in \mathbb{Z}^*$. Likewise, the error in the robot pose $\mathbf{x}_{r_k}^{(\lambda)}$ at time k is represented by the integer $\xi_{r_k}^{(\lambda)} \in \mathbb{Z}^*$. These integers store the path sizes (number of triangulations), and so they each provide a qualitative indication of the magnitude of a positional error. Note that if two landmarks have equal error values, this does not necessarily mean they have the same quantitative errors. It means they both share the same number of time instants where quantitative errors were accumulated.

At time $k = 0$, the initial landmarks in the map are each assigned an error value of zero, as they are based on zero triangulations. For time $k \geq 1$, the map of each hypothesis is updated using Algorithm 1. (Note that the hypothesis and time notation has been left out for brevity.) The variable ξ_S represents the error of each of the landmarks in scan S , along with the robot (i.e., $\xi_{r_k}^{(\lambda)} \leftarrow \xi_S^{(\lambda)}$). Each of the map landmarks \mathbf{x}_i has an associated index set T_i , which contains the indices of the landmarks used to triangulate it. For example, if landmark \mathbf{x}_{13} has the index set $T_{13} = \{3, 5, 8\}$, then it was triangulated using landmarks \mathbf{x}_3 , \mathbf{x}_5 and \mathbf{x}_8 . Of course, the index set associated with the initial landmarks

at time $k = 0$ is the null set \emptyset . The map landmarks used to triangulate the scan S is given by the index set T_S . Lastly, the correspondence set X represents all the landmark matches between the scan S and map M , as hypothesised by the data association algorithm.

Algorithm 1 Update_Map

Require: $k \geq 1$

- 1: $\xi_S \leftarrow \max\{\xi_i\}_{i \in T_S} + 1$
- 2: **for all** $\langle \mathbf{s}_a, \mathbf{x}_b \rangle \in X$ **do**
- 3: **if** $\xi_b \geq \xi_S$ **then**
- 4: $\mathbf{x}_b \leftarrow \mathbf{s}_a$ {in global coordinates}
- 5: $\xi_b \leftarrow \xi_S$
- 6: $T_b \leftarrow T_S$
- 7: **end if**
- 8: **end for**
- 9: add new landmarks with error ξ_S and triangulation T_S

The algorithm at time $k \geq 1$ basically operates as follows. First, the error ξ_S of the scan S is calculated using the map landmarks indexed in T_S . ξ_S is assigned the maximum error of these landmarks (a pessimistic approach) plus one, which accounts for one extra triangulation. Each of the landmarks in S that was matched in the data association process then replaces its corresponding map landmark, if it has the same or lower error value. The landmarks that were unmatched are added to the map, with reservations (as discussed in the previous subsection).

If the environment is static, as is commonly assumed, then this algorithm can be augmented in several ways. Firstly, any changes to a landmark \mathbf{x}_i can be recursively propagated down to all the landmarks whose position is correlated with it. To incorporate this, a function call "Propagate.Change(\mathbf{x}_b)" is inserted between lines 6 and 7 of Algorithm 1. The pseudo-code of this function is given in Algorithm 2. The benefit of this extension is an increase in the convergence rate; however, without any form of averaging, a moving landmark can potentially corrupt a large proportion of well-placed landmarks.

Another possible extension in a static environment is based on a time anomaly that occurs in the map building process. Without any dead-reckoning or motion continuity information, time does not play a role in ordering the scans or the way in which the map is build, as there is no causality. Therefore, the scans can be matched to the map in any order to maximise their data associations and, hence, their overlapping regions. Additionally, scans can be

Algorithm 2 Propagate.Change(\mathbf{x}_α)

- 1: **for** $i = 1$ to n **do**
- 2: **if** $\alpha \in T_i$ **then**
- 3: retriangulate \mathbf{x}_i
- 4: $\xi_i \leftarrow \max\{\xi_\beta\}_{\beta \in T_i} + 1$
- 5: Propagate.Change(\mathbf{x}_i) {recursive call}
- 6: **end if**
- 7: **end for**

continually rearranged to minimise the overall error of the resultant map. The number of scans that can be optimised in this manner, however, is limited by real-time constraints.

There are several issues about the proposed approach to SLAM that need to be considered. Firstly, the way in which the map converges is based on recursively triangulating each landmark's position using three of the most accurate landmarks that can be observed in the same scan. As a result, the map will converge to a limit that is determined by the landmark density and the sensor's range and accuracy. Particularly, the longer the range of the sensor, the more the map converges, and the lower the growth rate of positional errors outward from the origin. Note that the robot's navigational ability is not necessarily hampered when its far away from the origin, as its locally surrounding landmarks can have relative errors that are considerably lower than their global ones.

The well-known *loop closing problem* also needs to be considered, especially for cyclic environments [67, 102, 103, 104]. This problem occurs when the robot explores the environment, adding new landmarks with accumulating positional errors, and then returns to a place that it has been to before. The first issue that then needs to be resolved is the reliable detection of a previously visited place and, second, is the correction of accumulative error around the exploratory loop to maintain a consistent map. This correction is referred to as *closing the loop*. Generally, the first issue is difficult to resolve, as the robot's positional error can accumulate without bounds; however, if a previously visited place could be reliably detected, most SLAM methods are incapable of correcting errors backwards in time around the loop. To simplify matters, usually methods that are capable of closing the loop are limited to handling only small, well-shaped loops. A potential problem that exists, however, is that if the detection of a previously visited place is incorrect, then the process of closing the loop corrupts the map.

The first point to note about the proposed SLAM method is that the detection of a previously visited place is merely part of normal operation. There is also no odometric drift, and so compared to most of the other methods, the positional errors accumulate around the loop at a relatively small rate. The loop can be closed by, for example, reversing the triangulation order or processing the scans in a batch manner (as previously discussed); however, we argue against it. Closing the loop in such an explicit manner requires a decisive decision about the point of closure, which is always based on uncertain sensor information. While multiple map hypotheses can be maintained to address this issue, we suggest the use of another approach that is based on the map self-correcting over time. This approach involves removing landmark clusters from the map that are symmetric to other nearby clusters of significantly lower error. The result is a self-correcting phase where the loop is gradually replaced with a more accurate counterpart. By taking this passive approach, the loop can have an arbitrary shape (not size). However, during the time it takes for the

map to correct, as determined by the robot's navigational path, the robot may face an increase in the granularity of positional measurements.

D. Pathological Cases

There are several cases in which the proposed SLAM solution can fail. Firstly, since the solution is based entirely on exteroceptive sensing, it cannot function in an environment that has no detectable features. In essence, the system would become lost. Another failure case is a completely symmetrical environment (no distinguishing features). As an example, if the robot were to navigate down a straight road with only equidistant trees on either side, as depicted in Fig. 8, then the multiple hypotheses that are created cannot be resolved. Consequently, the SLAM process cannot determine if the robot is moving forward or differentiate between the robot standing still or driving at full speed. The unresolvable hypotheses can make it seem that the robot is teleporting between multiple poses, and for this reason, we call it the *teleportation problem*. Dead-reckoning information can be used to somewhat address this problem (discussed in the next subsection).

Failure can also occur in the case of a very large or overly cluttered environment. In these situations, the uniqueness of landmark relativities, such as inter-landmark distances, may diminish to a point where the number of possible data associations is too large to process in real-time. Also, if the map is too cluttered, then the tolerance bounds of the landmarks may permit an inordinate number of false data associations that appear to be symmetries. This can be counteracted by using a more exclusive landmark recognition technique; however, to sustain a large map, dead-reckoning information may need to be brought back into the equation.

E. Pulling in the Reins

So far, dead-reckoning information, motion continuity information and the robot's locomotive constraints have been purposely disregarded. Now, we shall briefly discuss how this information can be used to improve the performance of the proposed SLAM solution, without sacrificing its generality.

There are several ways in which dead-reckoning information from odometry and inertial sensors (gyroscopes, accelerometers, and combinations thereof, like an inertial measurement unit (IMU)) can be used to increase the efficiency and reliability of this solution, along with its ability to map large environments. For example, this information can be used to assist the data association process by adding weight to certain hypotheses or concentrating the data association search to only a localised portion of the map. In either case, the effects of odometric drift, especially when an inertial system is not used, is still avoided. Also, the robot's continuity of motion and locomotive constraints (e.g., its top speed) can be used to initiate the searches in the most likely area of the map and exclude pose hypotheses that are out of range.

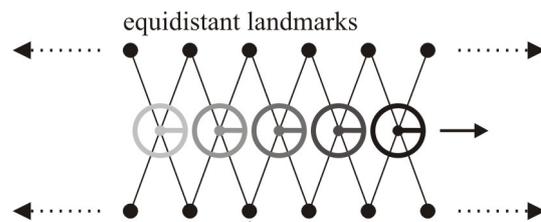


Fig. 8
THE TELEPORTATION PROBLEM

IV. CONCLUSIONS AND FUTURE WORK

This paper presented the first solution to SLAM that mimics the portability of a GPS receiver by purposely disregarding odometry and the assumption of continuity in the robot's motion. The solution functions by perpetually solving the kidnapped robot problem over time through sensing the local environment. This essentially decouples the SLAM process from the robot, and in doing so, there is no model of the robot or its physical interaction with the environment. Therefore, it can be implemented as a black-box system and transferred between robots, regardless of their physical makeup.

There are several other key benefits of this approach. Firstly, in comparison to the EKF, it is easy to implement; both in terms of the required academic competence of the roboticist and preparatory work (e.g., modeling time and effort). There are no stringent assumptions of the sensor errors, despite the sensors playing such a pivotal role. Also, the system is composed of simple generic elements, which tends to lead to a more enduring design than its rigid counterpart [105]. However, some may argue that this solution does not have an in-depth theoretical backing like the EKF. In rebuttal to this argument, the EKF's success on paper is primarily due to the rigidity of its assumptions, which hinder its real-world operation. Additionally, as a point of conjecture, the popularity of the EKF can be largely attributed to the undeniable ferocity at which it has been promoted.

While the proposed solution is unconventional, its universality warrants further investigation. Some of the many possible avenues for future work are listed below:

- **Landmark Recognition:** Enhancing the landmark recognition system is the most effective way of reducing data association ambiguities and increasing the maximum manageable map size. More landmark attributes can be detected, e.g., by fusing range and colour sensor data [94], or more redundancy can be used to reduce the likelihood of failure. Another possible extension is the use of landmark probabilities to represent the belief in their identification.
- **Complexity Analysis:** The data association method is implemented as an anytime algorithm and, hence, has a controllable run-time. However, a complexity

analysis would allow a better comparison with other methods, and can be used to make proactive judgments about the system load.

- Recovery Procedures: Possible recovery procedures need to be explored so that the robot can still navigate when a SLAM failure occurs. A graceful degradation scheme can be developed where the system operates at a reduced capacity while the global frame of reference is reestablished. Also, the system needs to be able to diagnose its own failure to obtain its extent.
- Dead-Reckoning Information: A comparative study to gauge the usefulness of odometry, especially in large environments, has not yet been performed.
- System Integration: There may be issues involved in integrating other navigational elements. For instance, the SLAM system builds a specialised map that may not show all the obstacles (e.g., a fallen log), which may be required by the path planner.
- 3D SLAM: Solving the kidnapped robot problem to additionally measure, with some degree of reliability, the robot's depth (or elevation), roll and pitch is a difficult proposition.
- Commercialisation: Ultimately, a standalone SLAM sensor can be made available off the shelf. Sensor specifications such as maximum robot speed, map size, and operating medium (outdoor, indoor, underwater etc.) can be used. The concept of miniaturisation also plays a role here.

REFERENCES

- [1] J. J. Leonard, P. M. Newman, R. J. Rikoski, J. Neira, and J. D. Tardós, "Towards robust data association and feature modeling for concurrent mapping and localization," in *10th Int. Symposium on Robotics Research*, Lorne, Victoria, Australia, 9-12 Nov. 2001, pp. 7–20.
- [2] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. Boston, MA: Academic Press, 1988.
- [3] I. J. Cox, "A review of statistical data association techniques for motion correspondence," *Int. J. Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.
- [4] R. A. Brooks, "Elephants don't play chess," *Robotics and Autonomous Systems*, vol. 6, no. 1-2, pp. 3–15, 1990.
- [5] D. J. Spero, "A review of outdoor robotics research," Dept. Electrical and Computer Systems Engineering, Monash University, Melbourne, Australia, Tech. Rep. MECSE-17-2004, 24 Nov. 2004. [Online]. Available: <http://www.ds.eng.monash.edu.au/techrep/reports/>
- [6] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.
- [7] S. P. Engelson and D. V. McDermott, "Error correction in mobile robot map learning," in *Proc. IEEE Int. Conf. Robotics and Automation*, Nice, France, 12-14 May 1992, pp. 2555–2560.
- [8] M. Yim, Y. Zhang, and D. Duff, "Modular robots," *IEEE Spectrum*, vol. 39, no. 2, pp. 30–34, 2002.
- [9] D. Rus, Z. Butler, K. Kotay, and M. Vona, "Self-reconfiguring robots," *Communications of the ACM*, vol. 45, no. 3, pp. 39–45, 2002.
- [10] R. R. Murphy, "Marsupial and shape-shifting robots for urban search and rescue," *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 14–19, 2000.
- [11] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: Sensors and techniques," *J. Robotic Systems*, vol. 14, no. 4, pp. 231–249, 1997.
- [12] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *4th Int. Symposium on Robotics Research*. MIT Press, 1987.
- [13] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. New York: Academic Press, 1979.
- [14] P. Moutarlier and R. Chatila, "Stochastic multi-sensory data fusion for mobile robot location and environment modelling," in *5th Int. Symposium on Robotics Research*, Tokyo, Japan, 28-31 Aug. 1989, pp. 85–94.
- [15] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems*, vol. 3, Osaka, Japan, 3-5 Nov. 1991, pp. 1442–1447.
- [16] S. B. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte, "Autonomous underwater simultaneous localisation and map building," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 24-28 Apr. 2000, pp. 1793–1798.
- [17] J. Guivant, E. Nebot, and S. Baiker, "Localization and map building using laser range sensors in outdoor applications," *J. Robotic Systems*, vol. 17, no. 10, pp. 565–583, 2000.
- [18] J. A. Castellanos, J. Neira, and J. D. Tardós, "Multi-sensor fusion for simultaneous localization and map building," *IEEE Trans. Robotics and Automation*, vol. 17, no. 6, pp. 908–914, 2001.
- [19] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, 2002.
- [20] S. J. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, 21-26 May 2001, pp. 4238–4243.

- [21] M. W. M. G. Dissanayake, H. F. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 2, San Francisco, CA, 24-28 Apr. 2000, pp. 1009–1014.
- [22] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Trans. Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.
- [23] K. S. Chong and L. Kleeman, "Feature-based mapping in real, large scale environments using an ultrasonic array," *Int. J. Robotics Research*, vol. 18, no. 1, pp. 3–19, 1999.
- [24] M. Csorba and H. F. Durrant-Whyte, "A new approach to map building using relative position estimates," in *Proc. SPIE: Navigation and Control Technologies for Unmanned Systems II*, Orlando, FL, 23 Apr. 1997, pp. 115–125.
- [25] J. K. Uhlmann, S. J. Julier, and M. Csorba, "Non-divergent simultaneous map-building and localization using covariance intersection," in *Proc. SPIE: Navigation and Control Technologies for Unmanned Systems II*, Orlando, FL, 23 Apr. 1997, pp. 2–11.
- [26] R. C. Luo and M. G. Kay, "Multisensor integration and fusion in intelligent systems," *IEEE Trans. Systems, Man and Cybernetics*, vol. 19, no. 5, pp. 901–931, 1989.
- [27] M. W. M. G. Dissanayake, P. Newman, H. F. Durrant-Whyte, S. Clark, and M. Csorba, "An experimental and theoretical investigation into simultaneous localisation and map building," in *Proc. Sixth Int. Symposium on Experimental Robotics*, Sydney, Australia, 26-28 Mar. 1999, pp. 265–274.
- [28] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
- [29] J. A. Castellanos, J. Neira, and J. D. Tardós, "Limits to the consistency of EKF-based SLAM," in *5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal, 5-7 Jul. 2004.
- [30] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," in *Proc. SPIE: Signal Processing, Sensor Fusion, and Target Recognition VI*, Orlando, FL, 21-24 Apr. 1997, pp. 182–193.
- [31] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robotics and Automation*, vol. 17, no. 6, pp. 890–897, 2001.
- [32] T. Bailey, E. M. Nebot, J. K. Rosenblatt, and H. F. Durrant-Whyte, "Data association for mobile robot navigation: A graph theoretic approach," in *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, San Francisco, CA, 24-28 Apr. 2000, pp. 2512–2517.
- [33] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automatic Control*, vol. AC-24, no. 6, pp. 843–854, 1979.
- [34] I. J. Cox and J. J. Leonard, "Probabilistic data association for dynamic world modeling: A multiple hypothesis approach," in *Fifth Int. Conf. Advanced Robotics*, Pisa, Italy, 19-22 Jun. 1991, pp. 1287–1294.
- [35] S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, vol. 21, no. 4, pp. 93–109, 2000.
- [36] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. Singapore: McGraw-Hill, 1991.
- [37] S. Thrun, "Robotic mapping: A survey," Dept. Computer Science, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-02-111, Feb. 2002.
- [38] S. Thrun, W. Burgard, and D. Fox, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Machine Learning*, vol. 31, no. 1-3, pp. 29–53, 1998.
- [39] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
- [40] G. J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 1997.
- [41] W. Burgard, D. Fox, D. Hennig, and T. Schmidt, "Estimating the absolute position of a mobile robot using position probability grids," in *Proc. Natl. Conf. Artificial Intelligence*, Portland, OR, 4-8 Aug. 1996, pp. 896–901.
- [42] A. Elfes, "Occupancy grids: A probabilistic framework for robot perception and navigation," Ph.D. dissertation, Dept. Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 1989.
- [43] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of the IEEE Conference on Robotics and Automation*, St. Louis, 1985, pp. 116–121.
- [44] C. Martin and S. Thrun, "Real-time acquisition of compact volumetric 3D maps with mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, 11-15 May 2002, pp. 311–316.
- [45] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. Natl. Conf. Artificial Intelligence*, Alta., Canada, 28 Jul.-1 Aug. 2002, pp. 593–598.
- [46] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F (Radar and Signal Processing)*, vol. 140, no. 2, pp. 107–113, 1993.
- [47] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.
- [48] S. Thrun, "Particle filters in robotics," in *Proc. 18th Conf. Uncertainty in Artificial Intelligence*, Alberta,

- Canada, 1-4 Aug. 2002, pp. 511–518.
- [49] D. B. Rubin, "Using the SIR algorithm to simulate posterior distributions," in *Bayesian Statistics 3*, J. M. Bernardo, M. H. DeGroot, D. V. Lindley, and A. F. M. Smith, Eds. Oxford, UK: Oxford University Press, 1988.
- [50] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall, 1986.
- [51] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Detroit, MI, 10-15 May 1999, pp. 1322–1328.
- [52] J. S. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Lausanne, Switzerland, 30 Sep.-5 Oct. 2002, pp. 454–459.
- [53] S. Zilberstein, "Resource-bounded sensing and planning in autonomous systems," *Autonomous Robots*, vol. 3, no. 1, pp. 31–48, 1996.
- [54] K. P. Murphy, "Bayesian map learning in dynamic environments," in *Neural Information Processing Systems*, Denver, CO, 29 Nov.-4 Dec. 1999, pp. 1015–1021.
- [55] G. Casella and C. P. Robert, "Rao-Blackwellisation of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81–94, 1996.
- [56] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "FastSLAM: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *J. Machine Learning Research*, 2004, to appear.
- [57] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Eighteenth Int. Joint Conf. Artificial Intelligence*, Acapulco, Mexico, 9-15 Aug. 2003, pp. 907–912.
- [58] M. Montemerlo, "FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2003.
- [59] O. King and D. A. Forsyth, "How does CONDENSATION behave with a finite number of samples?" in *6th Euro. Conf. Computer Vision*, Dublin, Ireland, 26 Jun.-1 Jul. 2000, pp. 695–709.
- [60] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," Dept. Engineering, Cambridge University, Cambridge, UK, Tech. Rep. CUED/F-INFENG/TR 380, 2000.
- [61] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*. Cambridge, MA: MIT Press, 1991.
- [62] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, 9-11 Apr. 1991, pp. 2724–2729.
- [63] P. J. Besl and H. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [64] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Third Int. Conf. 3-D Digital Imaging and Modeling*, Quebec City, Que., Canada, 28 May-1 Jun. 2001, pp. 145–152.
- [65] I. J. Cox, "Blanche - an experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. Robotics and Automation*, vol. 7, no. 2, pp. 193–204, 1991.
- [66] F. Lu and E. Milius, "Robot pose estimation in unknown environments by matching 2D range scans," *J. Intelligent and Robotic Systems*, vol. 18, no. 3, pp. 249–275, 1997.
- [67] J. S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, 8-9 Nov. 1999, pp. 318–325.
- [68] B. Jensen and R. Siegwart, "Scan alignment with probabilistic distance metric," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Sendai, Japan, 28 Sep.-2 Oct. 2004, pp. 2191–2196.
- [69] G. Weiss and E. V. Puttkamer, "A map based on laserscans without geometric interpretation," in *Proc. Int. Conf. Intelligent Autonomous Systems*, Karlsruhe, Germany, 27-30 Mar. 1995, pp. 403–407.
- [70] P. Biber, "The normal distributions transform: A new approach to laser scan matching," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, 27-31 Oct. 2003, pp. 2743–2748.
- [71] J. L. Crowley, F. Wallner, and B. Schiele, "Position estimation using principal components of range data," in *Proc. IEEE Int. Conf. Robotics and Automation*, Leuven, Belgium, 16-20 May 1998, pp. 3121–3128.
- [72] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "RoboCup: the robot world cup initiative," in *Proc. First Int. Conf. Autonomous Agents*, Marina del Rey, CA, 5-8 Feb. 1997, pp. 340–347.
- [73] J. S. Gutmann, T. Weigel, and B. Nebel, "Fast, accurate, and robust self-localization in polygonal environments," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Kyongju, South Korea, 17-21 Oct. 1999, pp. 1412–1419.
- [74] J. Weber, K. W. Jörg, and E. V. Puttkamer, "APR - global scan matching using anchor point relationships," in *6th Int. Conf. Intelligent Autonomous Systems*, Venice, Italy, 25-27 Jul. 2000, pp. 471–478.
- [75] M. Tomono, "A scan matching method using Euclidean invariant signature for global localization and map building," in *IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, 26 Apr.-1 May 2004,

- pp. 866–871.
- [76] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Las Vegas, NV, 27–31 Oct. 2003, pp. 206–211.
- [77] C. Pradalier and S. Sekhavat, "Concurrent matching, localization and map building using invariant features," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Lausanne, Switzerland, 30 Sep.–5 Oct. 2002, pp. 514–520.
- [78] P. M. Newman, "On the structure and solution of the simultaneous localisation and map building problem," Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, Australia, 1999.
- [79] E. C. Tolman, "Cognitive maps in rats and men," *The Psychological Review*, vol. 55, pp. 189–208, 1948.
- [80] J. Piaget and B. Inhelder, *The Child's Conception of Space*. New York: W. W. Norton, 1967.
- [81] A. W. Siegel and S. H. White, "The development of spatial representations of large-scale environments," in *Advances in Child Development and Behavior*, H. W. Reese, Ed. New York: Academic Press, 1975.
- [82] R. G. Golledge, Ed., *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. Baltimore, MD: Johns Hopkins University Press, 1999.
- [83] D. M. Mark, C. Freksa, S. C. Hirtle, R. Lloyd, and B. Tversky, "Cognitive models of geographical space," *Int. J. Geographical Information Science*, vol. 13, no. 8, pp. 747–774, 1999.
- [84] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, 1986.
- [85] T. S. Levitt and D. T. Lawton, "Qualitative navigation for mobile robots," *Artificial Intelligence*, vol. 44, no. 3, pp. 305–360, 1990.
- [86] B. Kuipers and Y. T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Robotics and Autonomous Systems*, vol. 8, no. 1–2, pp. 47–63, 1991.
- [87] D. J. Spero and R. A. Jarvis, "Towards exteroceptive based localisation," in *IEEE Conf. Robotics, Automation and Mechatronics*, Singapore, 1–3 Dec. 2004, pp. 822–827.
- [88] —, "On localising an unknown mobile robot," in *2nd Int. Conf. Autonomous Robots and Agents*, Palmerston North, New Zealand, 13–15 Dec. 2004, pp. 70–75.
- [89] L. G. Shapiro and R. M. Haralick, "Structural descriptions and inexact matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-3, no. 5, pp. 504–519, 1981.
- [90] H. R. Everett, *Sensors for Mobile Robots: Theory and Application*. Wellesley, MA: A. K. Peters, 1995.
- [91] R. A. Jarvis, "Range sensing for computer vision," in *Three-Dimensional Object Recognition Systems*, A. K. Jain and P. J. Flynn, Eds. Singapore: Elsevier, 1993.
- [92] M. Hebert, "Active and passive range sensing for robotics," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 24–28 Apr. 2000, pp. 102–110.
- [93] M. Kam, X. Zhu, and P. Kalata, "Sensor fusion for mobile robot navigation," *Proc. IEEE*, vol. 85, no. 1, pp. 110–119, 1997.
- [94] D. J. Spero and R. A. Jarvis, "3D vision for large-scale outdoor environments," in *Australasian Conf. Robotics and Automation*, Auckland, New Zealand, 27–28 Nov. 2002, pp. 228–233.
- [95] D. R. Chand and S. S. Kapur, "An algorithm for convex polytopes," *J. Association for Computing Machinery*, vol. 17, no. 1, pp. 78–86, 1970.
- [96] R. A. Jarvis, "On the identification of the convex hull of a finite set of points in the plane," *Information Processing Letters*, vol. 2, no. 1, pp. 18–21, 1973.
- [97] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," in *Computing in Euclidean Geometry*, D. Z. Du and F. Hwang, Eds. Singapore: World Scientific, 1992, pp. 23–90.
- [98] H. G. Barrow and R. M. Burstall, "Subgraph isomorphism, matching relational structures and maximal cliques," *Information Processing Letters*, vol. 4, no. 4, pp. 83–84, 1976.
- [99] P. M. Pardalos and J. Xue, "The maximum clique problem," *J. Global Optimization*, vol. 4, pp. 301–328, 1994.
- [100] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, Eds., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*. Chichester: Wiley, 1985.
- [101] A. Schrijver, "On the history of combinatorial optimization (till 1960)," in *Handbooks in Operations Research and Management Science*, K. Aardal, G. L. Nemhauser, and R. Weismantel, Eds. Amsterdam: Elsevier, 2005, vol. 12, ch. 1. [Online]. Available: <http://citeseer.ist.psu.edu/466877.html>
- [102] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots," in *IEEE Int. Conf. Robotics and Automation*, St. Louis, MO, 25–28 Mar. 1985, pp. 138–145.
- [103] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.
- [104] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco, CA, 24–28 Apr. 2000, pp. 321–328.
- [105] J. H. Saleh, "Perspectives in design: The Deacon's masterpiece and the hundred-year aircraft, spacecraft, and other complex engineering systems," *J. Mechanical Design*, 2005, to appear.