Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with Reversible Data Association

Charles Bibby
Department of Engineering Science
Oxford University
Email: cbibby@robots.ox.ac.uk

Ian Reid
Department of Engineering Science
Oxford University
Email: ian@robots.ox.ac.uk

Abstract— **The conventional technique for dealing with dynamic objects in SLAM is to detect them and then either treat them as outliers [20][1] or track them separately using traditional multi-target tracking [18]. We propose a technique that combines the least-squares formulation of SLAM and sliding window optimisation together with generalised expectation maximisation, to incorporate both dynamic and stationary objects directly into SLAM estimation. The sliding window allows us to postpone the commitment of model selection and data association decisions by delaying when they are marginalised permanently into the estimate. The two main contributions of this paper are thus: (i) using reversible model selection to include dynamic objects into SLAM and (ii) incorporating reversible data association. We show empirically that (i) if dynamic objects are present our method can include them in a single framework and hence maintain a consistent estimate and (ii) our estimator remains consistent when data association is difficult, for instance in the presence of clutter. We summarise the results of detailed and extensive tests of our method against various benchmark algorithms, showing its effectiveness.**

I. Introduction

SLAM in dynamic environments is essentially a model selection problem. The estimator constantly needs to answer the question: is a landmark moving or is it stationary? Although there are methods for doing model selection in recursive filtering frameworks such as interacting multiple model estimation or generalised pseudo-Bayesian estimation [9], these methods always have some lag before the model selection parameter(s) converge to the correct steady state. This means that for a period of time the filter could classify a target as dynamic when it is stationary or vice versa. This is potentially catastrophic for SLAM because incorrectly modeling a dynamic or stationary landmark will lead to biased measurements and hence map corruption and inconsistency.

We propose a framework that combines least-squares with sliding window optimisation [17] and generalised expectation maximisation [13]. This allows us to include reversible model selection and data association parameters in the estimation and hence include dynamic objects in the SLAM map robustly. At the heart of our method is the use of sliding window optimisation, which delays the point when information is marginalised out, allowing the filter a period of time to get the model selection and data association parameters correct before marginalisation. Although something similar could be achieved with a more traditional Extended Kalman Filter (EKF) using delayed decision making [11], the difference is that our method uses reversible as opposed to delayed decision making i.e. decisions can change many times in light of new information before being committed to the estimate.

The adverse effects of poor data association in SLAM, namely inconsistent estimates and divergence, are normally unacceptable and hence a suitable method must be selected. A common approach is the chi-squared nearest neighbour (NN) test, which assumes independence between landmarks and then probabilistically chooses the best measurement which falls within the gate of an individual landmark. This method can work well for sparsely distributed environments with good sensors; however, once the proximity between landmarks approaches the sensor noise or clutter is present, ambiguous situations arise and a more sophisticated method is required. One such method is joint compatibility branch and bound (JCBB) [14] which takes into account the correlations between landmarks by searching an interpretation tree [7] for the maximum number of jointly compatible associations. This method produces very good results when ambiguities are present, but still suffers from problems in the presence of clutter and is slow for large numbers of measurements. More recently the data association problem has been treated as a discrete optimisation over multiple time steps [15]. We also treat data association as a discrete optimisation, but include model selection and propose an alternative method that uses sliding window optimisation and generalised expectation maximisation [13](more specifically an approximate method called classification expectation maximisation [3]).

We begin by introducing our notation and the background on least-squares SLAM in Section II; Section III describes the background on sliding window optimisation; Section IV describes our method for doing SLAM with reversible data association; Section V extends this to SLAM in dynamic environments; Section VI compares our methods to an Iterated Extended Kalman Filter (IEKF) with either NN or JCBB for data association and finally in Section VII we conclude and discuss our ideas for future work.

II. Notation and least-squares SLAM

Although this section and the next cover background work [6][17], they have been included because they form the basis for our techniques.

A. Notation

- $\mathbf{x}_t$: A state vector describing the vehicle's pose (location and orientation $[x, y, \theta]$) at time t.
- $\mathbf{u}_t$: A control vector (odometry $[\dot{x_v}, \dot{y_v}, \dot{\theta_v}]$ in vehicle coordinates where $\dot{x_v}$ is in the direction the vehicle is pointing) that was applied to vehicle at time $t-1$ to take it to time t.
- $\mathbf{z}_t$: A measurement made by the vehicle at time t of a landmark in the world.
- $\mathbf{m}_k$: A state vector describing the location of landmark k.
- $\mathbf{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_t\}$: A set of vehicle poses.
- $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_t\}$: A set of odometry.
- $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$: A set of measurements.
- $\mathbf{M} = \{\mathbf{m}_0, \ldots, \mathbf{m}_k\}$: A set of all landmarks.

B. Least-Squares SLAM

Consider the Bayesian network in Figure 1 where each node represents a stochastic variable in the system. The grey nodes represent observed variables, the white nodes represent hidden variables and the arrows in the graph represent the dependence relationships between variables, for instance $\mathbf{z}_{t-1}$ depends upon $\mathbf{x}_{t-1}$ and $\mathbf{M}$. For the benefit of the reader let us reduce notation by making two assumptions: (i) only one observation per time step; (ii) we assume known data association i.e. which landmark generated a given measurement (we will relax this assumption from Section IV onwards).

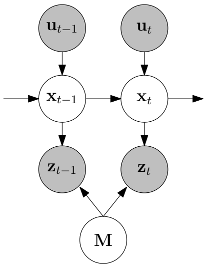

Fig. 1. A Bayesian network representing the SLAM problem.

The joint probability of $\mathbf{X}$, $\mathbf{M}$, $\mathbf{U}$ and $\mathbf{Z}$ can be factorised, using the independence structure depicted by the Bayesian network in Figure 1, as follows:

$$P(\mathbf{X}, \mathbf{M}, \mathbf{U}, \mathbf{Z}) = P(\mathbf{x}_0)P(\mathbf{M}) \times \prod_{t=1}^{T} P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{M})P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t). \quad (1)$$

where:

- T is the number of time steps.
- $P(\mathbf{x}_0)$ is the prior on vehicle state, which has a mean $\tilde{\mathbf{x}}_0$ and covariance $\mathbf{P}_0$.
- $P(\mathbf{M})$ is the prior on the map, which is normally taken to be the uninformative uniform distribution.
- $P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{M})$ is the measurement model i.e. the probability of the measurement $\mathbf{z}_t$ given the vehicle pose $\mathbf{x}_t$, the map $\mathbf{M}$ and the correct data association.
- $P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$ is the motion model i.e. the probability of the new pose $\mathbf{x}_t$ given the last vehicle pose $\mathbf{x}_{t-1}$ and the odometry $\mathbf{u}_t$.

If we take $P(\mathbf{M})$ to be the uninformative uniform distribution then (1) reduces to:

$$P(\mathbf{X}, \mathbf{M}, \mathbf{U}, \mathbf{Z}) = P(\mathbf{x}_0) \prod_{t=1}^{T} P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{M})P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t). \quad (2)$$

Let us now also make Gaussian assumptions and define the prior term, motion model and measurement model respectively as:

$$\mathbf{x}_0 = \tilde{\mathbf{x}}_0 + \mathbf{p}_0 \;\Leftrightarrow\; P(\mathbf{x}_0) \propto exp(-\frac{1}{2}\left\|\tilde{\mathbf{x}}_0 - \mathbf{x}_0\right\|^2_{\mathbf{P}_0}) \quad (3)$$

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{q}_t \;\Leftrightarrow\; P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) \propto exp(-\frac{1}{2}\left\|f(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\right\|^2_{\mathbf{Q}_t}) \quad (4)$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{M}) + \mathbf{r}_t \;\Leftrightarrow\; P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{M}) \propto exp(-\frac{1}{2}\left\|h(\mathbf{x}_t, \mathbf{M}) - \mathbf{z}_t\right\|^2_{\mathbf{R}_t}) \quad (5)$$

where $\mathbf{p}_0$, $\mathbf{q}_t$ and $\mathbf{r}_t$ are normally distributed, zero mean, noise vectors with covariances $\mathbf{P}_0$, $\mathbf{Q}_t$ and $\mathbf{R}_t$ respectively. The $\|\mathbf{e}\|^2_{\mathbf{\Sigma}}$ notation represents the squared Mahalanobis distance $\mathbf{e}^T\mathbf{\Sigma}^{-1}\mathbf{e}$, where $\mathbf{\Sigma}$ is a covariance. We can now perform inference on the Bayesian network in Figure 1 to find the maximum a posteriori (MAP) estimate $\{\hat{\mathbf{X}}, \hat{\mathbf{M}}\} = \arg\max_{\{\mathbf{X},\mathbf{M}\}} P(\mathbf{X}, \mathbf{M}|\mathbf{U}, \mathbf{Z})$. This can be done by minimising the negative log of the joint distribution (2):

$$\{\hat{\mathbf{X}}, \hat{\mathbf{M}}\} \triangleq \underset{\{\mathbf{X},\mathbf{M}\}}{\arg\min}\,(-\log(P(\mathbf{X}, \mathbf{M}, \mathbf{U}, \mathbf{Z}))). \quad (6)$$

By substituting Equations (3), (4) and (5) into (6) we get a non-linear least-squares problem of the form:

$$\{\hat{\mathbf{X}}, \hat{\mathbf{M}}\} \triangleq \underset{\{\mathbf{X},\mathbf{M}\}}{\arg\min} \Big\{ \|\tilde{\mathbf{x}}_0 - \mathbf{x}_0\|^2_{\mathbf{P}_0} + \sum_{t=1}^{T}\left(\|f(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\mathbf{Q}_t} + \|h(\mathbf{x}_t, \mathbf{M}) - \mathbf{z}_t\|^2_{\mathbf{R}_t}\right)\Big\}. \quad (7)$$

Let us now linearise the non-linear terms and re-write as a matrix equation:

$$\{\hat{\mathbf{X}}, \hat{\mathbf{M}}\} \triangleq \underset{\{\mathbf{X},\mathbf{M}\}}{\arg\min} \Big\{ \|\mathbf{B}\boldsymbol{\delta}\mathbf{x}_0 - \{\mathbf{x}_0 - \tilde{\mathbf{x}}_0\}\|^2_{\mathbf{P}_0} + \sum_{t=1}^{T} \big(\|\{\mathbf{F}_{t-1}\boldsymbol{\delta}\mathbf{x}_{t-1} + \mathbf{B}\boldsymbol{\delta}\mathbf{x}_t\} - \{\mathbf{x}_t - f(\mathbf{x}_{t-1}, \mathbf{u}_t)\}\|^2_{\mathbf{Q}_t} + \|\{\mathbf{H}_t\boldsymbol{\delta}\mathbf{x}_t + \mathbf{J}_t\boldsymbol{\delta}\mathbf{M}\} - \{\mathbf{z}_t - h(\mathbf{x}_t, \mathbf{M})\}\|^2_{\mathbf{R}_t} \big) \Big\}, \quad (8)$$

where $\mathbf{F}_{t-1}$ is the Jacobian of $f(.)$ w.r.t. $\mathbf{x}_{t-1}$, $\mathbf{H}_t$ is the Jacobian $h(.)$ w.r.t. $\mathbf{x}_t$ and $\mathbf{J}_t$ is the Jacobian of $h(.)$ w.r.t. $\mathbf{M}$. We have also introduced $\mathbf{B} = -\mathbf{I}_{d\times d}$ so that $\boldsymbol{\delta}\mathbf{x}_t$ which is a small change in the states corresponding to the pose at time t is treated in the same way as the other terms; where d is the dimension of a single vehicle pose.

We can now factorise and write a standard least-squares matrix equation:

$$\mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A}\boldsymbol{\delta} = \mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{b}, \quad (9)$$

where $\mathbf{A}$ is a matrix of Jacobians, $\boldsymbol{\Sigma}$ is a covariance matrix and $\mathbf{b}$ is an error vector, for a detailed look at the structure of these matrices refer to [6]. We solve for $\boldsymbol{\delta}$ in (9) using direct sparse methods [4][6].

C. The Hessian, Information Matrix and Inverse Covariance

If the Cramer Rao lower bound [16] is reached, then given that we have normally distributed zero mean variables the following condition is satisfied:

$$\mathbf{P}^{-1} = \mathbf{Y} = \mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A}$$

where $\mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A}$ is the approximation to the Hessian calculated when solving the least-squares problem (9). This is why the information matrix $\mathbf{Y}$ in the information filter version of SLAM [19] and the Hessian in the least-squares formulation of SLAM [17] are equivalent to the inverse of the covariance matrix $\mathbf{P}$ in the more traditional Kalman filter based SLAM systems. Interestingly, as explained fully in [6], the non-zero elements of the information matrix $\mathbf{Y}$ correspond to links in the Markov Random Field (MRF) that is equivalent to the Bayesian network in Figure 1. Each of these links represents a constraint or relationship between two nodes in the MRF, e.g. a measurement equation linking a vehicle pose to a landmark or an odometry equation linking one vehicle pose to the next. The first row of Figure 2 shows the structure of the information matrix and MRF for a simple 2D example. Let us now consider the structure of the information matrix in the first row of Figure 2: Y_v is block tridiagonal and represents the information from odometry between vehicle poses; Y_m is block diagonal and represents the information about landmarks in the map and Y_{vm} and Y_{vm}^T represent the information associated with measuring a landmark from a given pose. We will revisit this simple example in Section III-A when discussing how marginalisation affects the structure of both $\mathbf{Y}$ and the MRF.

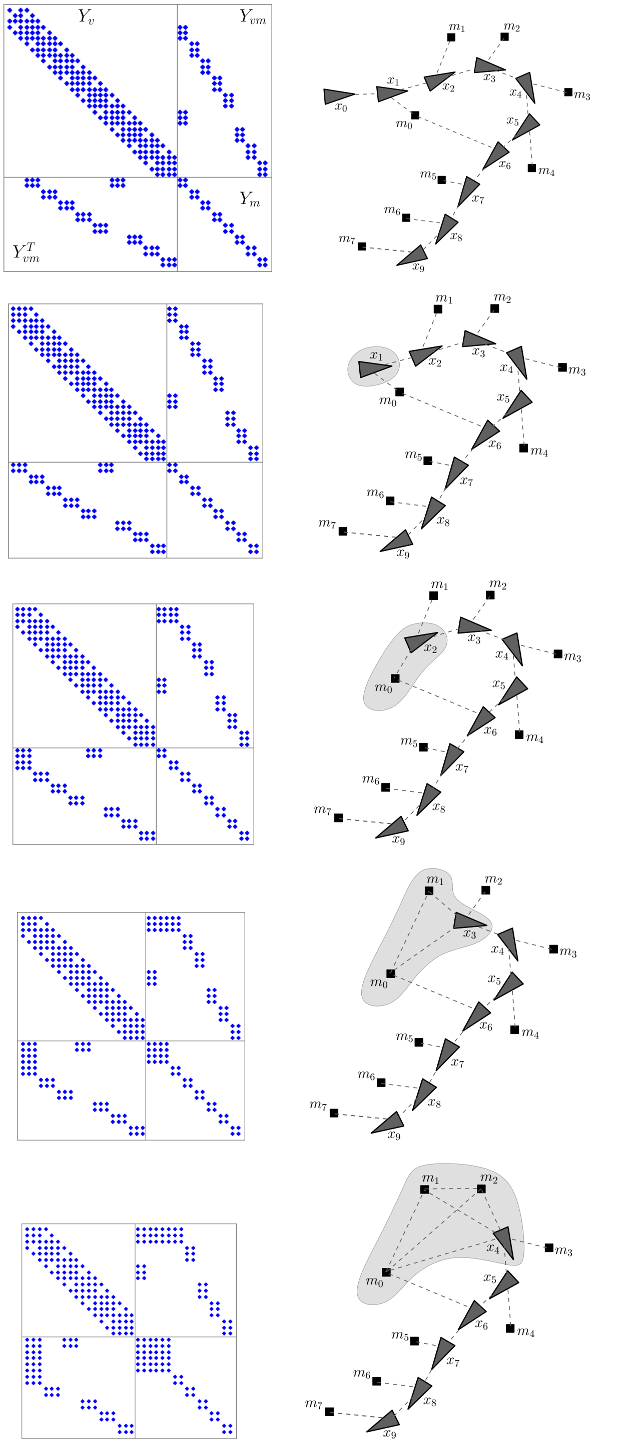

Fig. 2. Illustrates how the information matrix (left column) and MRF (right column) change as poses $\mathbf{x}_0$ (row 2), $\mathbf{x}_1$ (row 3), $\mathbf{x}_2$ (row 4) and $\mathbf{x}_3$ (row 5) are marginalised out. The light grey region indicates the Markov blanket that corresponds to the prior term (11) in Equation (10).

III. Sliding Window SLAM

Each iteration of "standard" EKF-based SLAM provides a maximum a posteriori estimate for the state at the current time step. Modern understanding of this recognises that previous poses have been marginalised out. In contrast, Full-SLAM [2] finds the maximum a posteriori estimate of the entire pose history. This is akin to bundle-adjustment techniques from photogrammetry [8], and has the advantage that more accurate solutions can be found since optimisation is performed over past and future data. It does of course suffer from the problem of growth without bound in the state size. Recently [17] proposed *Sliding Window SLAM*; where optimisation is carried out over a time window of length τ. This aims to capture the advantages of Full-SLAM, but rather than retaining the entire trajectory history, poses older than τ are marginalised out. In our work we take advantage of the optimisation over the trajectory history not only to improve the pose estimates, but crucially in order to allow reversible data association and model selection to take place. Details of this are presented in section IV, but first we review the two governing equations of sliding window SLAM which are: (i) an optimisation step

$$\{\hat{\mathbf{x}}_{t-\tau:t}, \hat{\mathbf{M}}\} = \underset{\{\mathbf{x}_{t-\tau:t},\mathbf{M}\}}{\arg\max} \Bigg(P(\mathbf{x}_{t-\tau}, \mathbf{M}|\mathbf{z}_{1:t-\tau}, \mathbf{u}_{1:t-\tau}) \times \prod_{j=t-\tau+1}^{t} P(\mathbf{z}_j|\mathbf{x}_j, \mathbf{M}) P(\mathbf{x}_j|\mathbf{x}_{j-1}, \mathbf{u}_j) \Bigg) \tag{10}$$

and (ii) a marginalisation step

$$P(\mathbf{x}_{t-\tau+1}, \mathbf{M}|\mathbf{z}_{1:t-\tau+1}, \mathbf{u}_{1:t-\tau+1}) = \int P(\mathbf{x}_{t-\tau+1}, \mathbf{x}_{t-\tau}, \mathbf{M}|\mathbf{z}_{1:t-\tau+1}, \mathbf{u}_{1:t-\tau+1}) \mathbf{d}\mathbf{x}_{t-\tau}. \tag{11}$$

The term $P(\mathbf{x}_{t-\tau}, \mathbf{M}|\mathbf{z}_{1:t-\tau}, \mathbf{u}_{1:t-\tau})$ is the prior used at time t and is just the posterior at time $t-\tau$ i.e. the distribution of vehicle pose and map at the beginning of the sliding window. In practice this is only recalculated when $t-\tau > 0$, before that time the distribution of initial vehicle pose $P(\mathbf{x}_0)$ is used and the marginalisation step is left out.

A. Marginalisation in Information Form

It is well known that it is possible to decouple states $\mathbf{y}_1$ from a system of equations of the form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix} \tag{12}$$

using the Schur Complement [8] method. The idea is to pre multiply both sides of the equation with the matrix $[\mathbf{I} \quad \mathbf{0}; -\mathbf{B}^T\mathbf{A}^{-1} \quad \mathbf{I}]$, which results in a system of equations where $\mathbf{y}_2$ can be solved independently of $\mathbf{y}_1$ i.e.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{b}_1 \end{bmatrix} \tag{13}$$

The term $\mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ is known as the Schur Complement and corresponds to the information matrix for the decoupled system. If this system of equations represents a least-squares problem as described in Section II then this is equivalent to marginalising out the random variables $\mathbf{y}_1$. Let us now consider what happens to the structure of $\mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ as old poses are marginalised out. Figure 2 shows the effect of marginalising out poses one by one. The first row shows the situation before any marginalisation. The second row corresponds to marginalising out $\mathbf{x}_0$, which results in no change of structure in the information matrix (because no features were observed from this pose) but does introduce a prior on the vehicle state $\mathbf{x}_1$. Then in the third row $\mathbf{x}_1$ has been marginalised out and a link has been introduced between $\mathbf{x}_2$ and $\mathbf{m}_0$, which is also seen in the $\mathbf{Y}_{vm}$ block of the information matrix (this extra link and the prior on $\mathbf{x}_2$ and $\mathbf{m}_0$ is explained by the prior term in Equation (10)). As poses $\mathbf{x}_2$ and $\mathbf{x}_3$ are marginalised out more links are again introduced between the oldest pose and the landmarks; links are also introduced between landmarks that are no longer observed from the oldest pose. In practice we use the prior term (3) in our least-squares formulation to represent the prior term (11) in the sliding window; where we replace $\tilde{\mathbf{x}}_0$ with $\mathbf{y}_2$, $\mathbf{x}_0$ with the poses and landmarks that have a prior and $\mathbf{P}_0$ with $(\mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B})^{-1}$. *To maintain probabilistic correctness only equations containing the pose being marginalised out should be included in the system* (12) *and then the modified system* (13) *should be solved for* $\mathbf{y}_2$.

IV. Reversible Data Association

So far the previous sections have covered the necessary background knowledge, let us now introduce the first of our two algorithms. We first relax our assumption of known data association and introduce integer data association parameters $\mathbf{D} \triangleq \{d_1, \ldots, d_t\}$, which assign measurement $\mathbf{z}_t$ to landmark $\mathbf{m}_{d_t}$. By combining sliding window estimation and least-squares with generalised expectation maximisation we can estimate both the continuous state estimates $\{\hat{\mathbf{X}}, \hat{\mathbf{M}}\}$ and the discrete data association parameters $\mathbf{D}$.

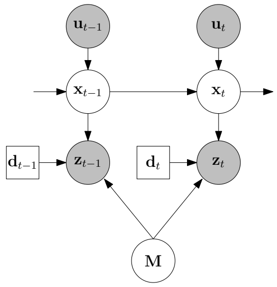

Fig. 3. A Bayesian network representing SLAM with reversible data association (Note:- Square boxes indicate discrete variables).

Figure 3 illustrates the Bayesian network that corresponds to the joint distribution of the relaxed problem:

$$P(\mathbf{X}, \mathbf{M}, \mathbf{D}, \mathbf{U}, \mathbf{Z}) = \tag{14}$$

$$P(\mathbf{x}_0)P(\mathbf{M})P(\mathbf{D}) \prod_{t=1}^{T} P(\mathbf{z}_t|\mathbf{x}_t, \mathbf{M}, d_t) P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) \quad . \tag{15}$$

What we are really after is the MAP estimate $P(\mathbf{X},\mathbf{M}|\mathbf{U},\mathbf{Z})$, whereas what we have is $P(\mathbf{X},\mathbf{M},\mathbf{D},\mathbf{U},\mathbf{Z})$ where $\mathbf{D}$ is considered a nuisance parameter. One solution would be to marginalise $\mathbf{D}$ out i.e.

$$P(\mathbf{X},\mathbf{M}|\mathbf{U},\mathbf{Z}) = \int_{\mathbf{D}} P(\mathbf{X},\mathbf{M},\mathbf{D}|\mathbf{U},\mathbf{Z})\mathbf{dD}.$$

Unfortunately in practice this integral is computationally intractable because the number of permutations of $\mathbf{D}$ grows exponentially with the length of the sliding window. A more tractable solution is to use the expectation maximisation algorithm [10], [5] to estimate $P(\mathbf{X},\mathbf{M}|\mathbf{U},\mathbf{Z})$. If we let $\boldsymbol{\Theta} = \{\mathbf{X},\mathbf{M}\}$ and $\boldsymbol{\Psi} = \{\mathbf{U},\mathbf{Z}\}$ then we would like $P(\boldsymbol{\Theta}|\boldsymbol{\Psi})$ as opposed to $P(\boldsymbol{\Theta},\mathbf{D}|\boldsymbol{\Psi})$. The expectation maximisation algorithm achieves this by recursively applying the following two steps:

- E-Step: calculate $P(\mathbf{D}|\boldsymbol{\Theta}^k,\boldsymbol{\Psi})$.
- M-Step: $\boldsymbol{\Theta}^{k+1} = \arg\max_{\boldsymbol{\Theta}} \left(\int_{\mathbf{D}} P(\mathbf{D}|\boldsymbol{\Theta}^k,\boldsymbol{\Psi})\right.$ $\left.\log P(\boldsymbol{\Theta}|\mathbf{D},\boldsymbol{\Psi})\mathbf{dD}\right)$.

A common simplification that is often applied to make the M-Step even more tractable is the ‘winner-take-all’ approach also known as classification expectation maximisation [3], [12], which assumes $P(\mathbf{D}|\boldsymbol{\Theta}^k,\boldsymbol{\Psi})$ to be a delta function centered on the best value of $\mathbf{D}$, reducing the algorithm to:

- E-Step: $\mathbf{D}^{k+1} = \arg\max_{\mathbf{D}} P(\mathbf{D}|\boldsymbol{\Theta}^k,\boldsymbol{\Psi})$.
- M-Step: $\boldsymbol{\Theta}^{k+1} = \arg\max_{\boldsymbol{\Theta}} P(\boldsymbol{\Theta}|\mathbf{D}^{k+1},\boldsymbol{\Psi})$.

Finally, it has been shown that it is not necessary to complete the maximisation but that a single step where $P(\boldsymbol{\Theta}^{k+1}|\mathbf{D}^{k+1},\boldsymbol{\Psi}) >= P(\boldsymbol{\Theta}^{k}|\mathbf{D}^{k},\boldsymbol{\Psi})$ is not only sufficient for convergence but often improves the rate of convergence [13]. For probabilistic correctness it is necessary to use the joint distribution over landmarks and a single pose during the E-Step i.e. JCBB. In practice this is very slow for large numbers of measurements and so we also include in our results an implementation which makes an extra assumption of landmark independence during the E-Step i.e. chi-squared NN. In practice this method gives a significant improvement over other methods (which do not use reversible data association) without the full cost of JCBB. It is also interesting at this point to draw on the similarity between this approach and iterative closest point; the significant difference is that our method uses the underlying probability distribution (Mahalanobis distances) to find the most likely correspondences as opposed to the closest in a euclidean sense. Algorithm 1 gives a summary of our method.

V. SLAM IN DYNAMIC ENVIRONMENTS

We will now introduce our method for SLAM in dynamic environments (SLAMIDE). Let us start by relaxing the problem even further by: (i) introducing model selection parameters $\mathbf{V}_T \triangleq \{v_T^0,\ldots,v_T^k\}$, which consist of a binary indicator variable per landmark taking the value **stationary** or **dynamic** with probability p, $1-p$ respectively; (ii) extending the state vector for each landmark to include velocities $\dot{\mathbf{x}}$, $\dot{\mathbf{y}}$ and (iii) using the estimated landmark velocities as observations to a Hidden Markov Model (HMM) that estimates the probability of an object being dynamic or stationary. Figure 4 is a Bayesian network that shows our formulation of the SLAMIDE problem, where the most significant changes from normal SLAM are:

- The map becomes time dependent $\mathbf{M} \triangleq \{\mathbf{M}_0,\ldots,\mathbf{M}_t\}$.
- Model selection parameters $\mathbf{V}_T \triangleq \{v_T^0,\ldots,v_T^k\}$ are introduced.
- Data association parameters $\mathbf{D} \triangleq \{d_1,\ldots,d_t\}$ are introduced.

Algorithm 1: SLAM with reversible data association.

```
$\mathbf{P} = \mathbf{P}_0; \mathbf{x}_0 = \tilde{\mathbf{x}}_0; \mathbf{M} = []; \mathbf{D} = [];$
for t=[0:T] do
    DoVehiclePrediction();
    while $|\boldsymbol{\delta}|_\infty > \epsilon$ do
        $\hat{\mathbf{D}}$ = DoDataAssociation();
        AddAnyNewLandmarks();
        Compute $\mathbf{A}$, $\boldsymbol{\Sigma}$ and $\mathbf{b}$;
        Solve for $\boldsymbol{\delta}$ in $\mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A}\boldsymbol{\delta} = \mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{b}$;
        $\{\hat{\mathbf{x}}_{t-\tau:t}, \hat{\mathbf{M}}\} = \{\hat{\mathbf{x}}_{t-\tau:t}, \hat{\mathbf{M}}\} + \boldsymbol{\delta}$;
        Compute $\mathbf{P}$ using triangular solve;
    end
    if $t - \tau > 0$ then
        Compute $\mathbf{y}_2$ and $\mathbf{D} - \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ using Schur
        Complement method (see Section III-A);
    end
end
```

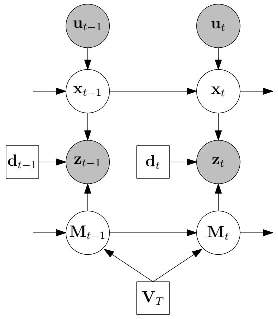

Fig. 4. A Bayesian network representing SLAMIDE (Note:- Square boxes indicate discrete variables).

The corresponding joint distribution $P(\mathbf{X},\mathbf{M},\mathbf{D},\mathbf{V},\mathbf{U},\mathbf{Z})$ from Figure 4 is:

$$P(\mathbf{X},\mathbf{M},\mathbf{D},\mathbf{V},\mathbf{U},\mathbf{Z}) = P(\mathbf{x}_0)P(\mathbf{M}_0)P(\mathbf{D})P(\mathbf{V}_T)\times \prod_{t=1}^{T} P(\mathbf{z}_t|\mathbf{x}_t,\mathbf{M}_t,d_t)P(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)P(\mathbf{M}_t|\mathbf{M}_{t-1},\mathbf{V}_T), \tag{16}$$

where:

- $P(\mathbf{V}_T)$ is a prior on the model selection parameters.

- $P(\mathbf{M}_t|\mathbf{M}_{t-1}, \mathbf{V}_T)$ is the motion model for the map given the current estimate of the model selection parameters. We use constant position for stationary landmarks and constant velocity with noise in $\dot{x}$ and $\dot{y}$ for dynamic landmarks.

Following the same principles used in our reversible data association method and including the extra nuisance parameter $\mathbf{V}$ we propose the following five steps to solve the optimisation:

1) $\mathbf{D}^{k+1} = \arg\max_{\mathbf{D}} P(\mathbf{D}|\mathbf{\Theta}^k, \mathbf{V}^k, \mathbf{\Psi})$
2) $\mathbf{\Theta}^{k+1} = \arg\max_{\mathbf{\Theta}} P(\mathbf{\Theta}|\mathbf{D}^{k+1}, \mathbf{V}^k, \mathbf{\Psi})$
3) $\mathbf{M}^{k+1'} = \arg\max_{\mathbf{M}} P(\mathbf{M}|\mathbf{X}^{k+1}, \mathbf{D}^{k+1}, \mathbf{V} = \mathbf{dyn}, \mathbf{\Psi})$
4) $\mathbf{V}^{k+1} = \arg\max_{\mathbf{V}} P(\mathbf{V}|\mathbf{X}^{k+1}, \mathbf{M}^{k+1'}, \mathbf{D}^{k+1}, \mathbf{\Psi})$
5) $\mathbf{\Theta}^{k+1} = \arg\max_{\mathbf{\Theta}} P(\mathbf{\Theta}|\mathbf{D}^{k+1}, \mathbf{V}^{k+1}, \mathbf{\Psi})$

Step 1: performs the data association using either NN or JCBB. In practice this is actually also computed at every iteration in steps 2, 3 and 5.

Step 2: is a least-squares optimisation for the vehicle poses and landmark states using the new data association. The main purpose of this optimisation is to refine the predicted vehicle and landmark locations using the new measurements. In practice this step is particularly important if the vehicle prediction is poor (large odometry noise), because large vehicle uncertainty gives rise to an ambiguous situation where it is hard to differentiate between vehicle and landmark motion.

Step 3: optimises for the landmark states assuming all landmarks are dynamic whilst holding the vehicle poses constant. The reason the vehicle poses are held constant is to remove any ambiguity between vehicle and landmark motion. This is reasonable if most of the landmarks maintain their model selection between time steps and hence the $\mathbf{X}^{k+1}$ given by Step 2 is close to the optimal answer.

Step 4: takes the answer from Step 3 and computes the next step of the HMM using a recursive Bayesian filter; where the likelihood model is a Gaussian on the average velocity with σ=2.0m/s and μ=0 and the prior $P(\mathbf{V}_T)$ for landmark j is:

$$P(v_T^j = \mathbf{stationary}) = \begin{cases} 0.6 & \text{if } v_{T-1}^j = \mathbf{stationary}, \\ 0.4 & \text{if } v_{T-1}^j = \mathbf{dynamic}, \end{cases}$$

which is based on $\mathbf{V}_{T-1}$ the model selection parameters chosen at the last time step. Given that the probability $P(v_T^j = \mathbf{stationary}) = p$ and $P(v_T^j = \mathbf{dynamic}) = 1 - p$ we threshold at 0.5 to get a discrete decision on which model to use.

Step 5: is a least-squares optimisation with the new model selection and data association parameters (using the answer from Step 2 as the starting point for optimisation). This step refines the estimate from Step 2 taking into account any changes in model selection to give the final estimate for this time step.

In practice this whole process only requires a few least-squares iterations typically: two in Step 2; one in Step 3 and two or three in Step 5; Step 1 and Step 4 are solved directly.

Map Management: When adding a new landmark we initialise its model selection probability to 0.5 (to reflect the uncertainty in whether it is dynamic or stationary) and add a very weak prior of zero initial velocity; this weak prior is essential to make sure that a landmark's velocity is always observable and hence our system of equations is positive definite. We also remove any dynamic landmarks that are not observed within the sliding window, this is done for two reasons: (i) real world objects do not obey a motion model exactly and so errors accumulate if you predict for too long and (ii) if you continue predicting a dynamic landmark and hence adding noise, then at some point measurements begin to get incorrectly associated to it due to the Mahalanobis test.

VI. Results

We use two simple 2D environments, which cover 400m by 400m, one with 15 landmarks and the other with 20 landmarks. In both environments the vehicle moves between three waypoints at 5m/s using proportional heading control (max. yaw rate 5°/sec) and provides rate-of-turn, forwards velocity and slip as odometry with covariance $\mathbf{Q}$. It has a range bearing sensor with a 360° field-of-view, 400m range and zero mean Gaussian noise added with covariance $\mathbf{R}$. The second environment is used for the dynamic object experiment where we progressively change stationary landmarks to dynamic landmarks, which move between waypoints using the same control scheme, speed and rate-of-turn as the vehicle.

We compare our method using either NN or JCBB for data association against an IEKF with either NN or JCBB. All experiments are 60 time steps long and use: a chi-squared threshold of $\boldsymbol{v}^T\mathbf{S}^{-1}\boldsymbol{v} < 16$; a sliding window length of 6 time steps; odometry noise (all noise quotes are for 1σ) of 0.1m/s on forwards velocity and 0.01m/s on slip; measurement noise of 1m for range and 0.5° for bearing; a maximum number of 8 iterations; the same stopping condition and the same initial vehicle uncertainty. The reason we use such a large chi-squared threshold is because for any significant angular uncertainty linearising the prediction covariance can cause all measurements to fall outside their data association gates.

In order to compare the performance of the algorithms we use two metrics: (i) the percentage of correct data association, which we define to be the ratio of correct associations between time steps w.r.t the number of potential correct associations (×100) and (ii) the percentage of consistent runs where we use the following test to determine whether a run is consistent: compute the Normalised Estimation Error Squared (NEES), which is defined as $D_t^2 = (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \mathbf{P}_t^{-1} (\mathbf{x}_t - \hat{\mathbf{x}}_t)$ and then for each time step perform the corresponding chi-squared test $D_t^2 \leq \chi^2_{r,1-\alpha}$ where r is the dimension of $\mathbf{x}_t$ and α is a threshold. We take the threshold α to be 0.05 and can then compute the probability of this test failing k times out of n from the binomial distribution $B(n, \alpha)$, which we use to threshold on the number of times the test can fail before we are 99% certain that a run is inconsistent.

We have carried out three Monte-Carlo simulation experiments (where each point on the graphs has been generated from 100 runs):

Figure 5 - Noise in rate-of-turn odometry: Performance was tested without clutter against increasing noise in rate-of-turn odometry with a variance of 1° to 60°. The IEKFJCBB and our RDJCBB both perform perfectly with data association but start becoming inconsistent more often for higher noise levels. As expected our RDNN outperforms the IEKFNN and matches the performance of IEKFJCBB up to around 25° of noise; this is interesting because it shows the RDNN could be used as a faster alternative to IEKFJCBB for medium noise problems.

Figure 6 - Number of clutter measurements: Performance was tested with a noise of 1° for rate-of-turn odometry against increasing clutter from 0 to 100 clutter measurements within the sensor range. This is where the real benefit of reversible data association becomes apparent. All algorithms tested use the same map management scheme, which is to remove landmarks that are not observed for three consecutive time steps after they have been added to the map. In the traditional IEKF this is done by simply removing them from the state vector and covariance matrix (marginalisation); whereas with our scheme if the information is removed before marginalisation i.e. the sliding window is longer than the time required to carry out map management then there is no effect on the estimate. This is clear from Figure 6 as both of our methods maintain their consistency with increasing clutter as opposed to the IEKF based methods which tail off.

Figure 7 - Percentage of dynamic objects: Performance was tested without clutter and with a noise of 1° for rate-of-turn odometry against an increasing percentage of dynamic objects from 0 to 100 percent. The figure clearly shows that using SLAMIDE to include dynamic objects allows us to navigate in regions with dynamic objects. We maintain a good level of consistency up to 90% of dynamic objects at which point the performance degrades until at 100% every run is inconsistent, which is because the system is no longer observable; i.e. there are ambiguities between vehicle and landmark motion.

Timing Results: With 20 measurements per time step and a sliding window length of 6 time steps on a 3.6GHz Pentium 4 the IEKF and SLAM with reversible data association run at approximately 30Hz and SLAMIDE runs at about 3Hz. We believe this can be significantly improved upon as we have yet to fully optimise the code, for instance we currently do a dense solve for $\mathbf{P} = \mathbf{Y}^{-1}$ which is a bottleneck (this could be heavily optimised or possibly avoided completely). Also, once a landmark has been created and passed the map management test, it always remains in the estimate; however, sliding window estimation is constant time if you choose to marginalise out landmarks i.e. maintain a constant state size.

VII. Conclusions and Future Work

We have proposed a method that combines sliding window optimisation and least-squares together with generalised expectation maximisation to do reversible model selection and data association. This allows us to include dynamic objects directly into the SLAM estimate, as opposed to other techniques which typically detect dynamic objects and then either treat them as outliers [20][1] or track them separately [18].

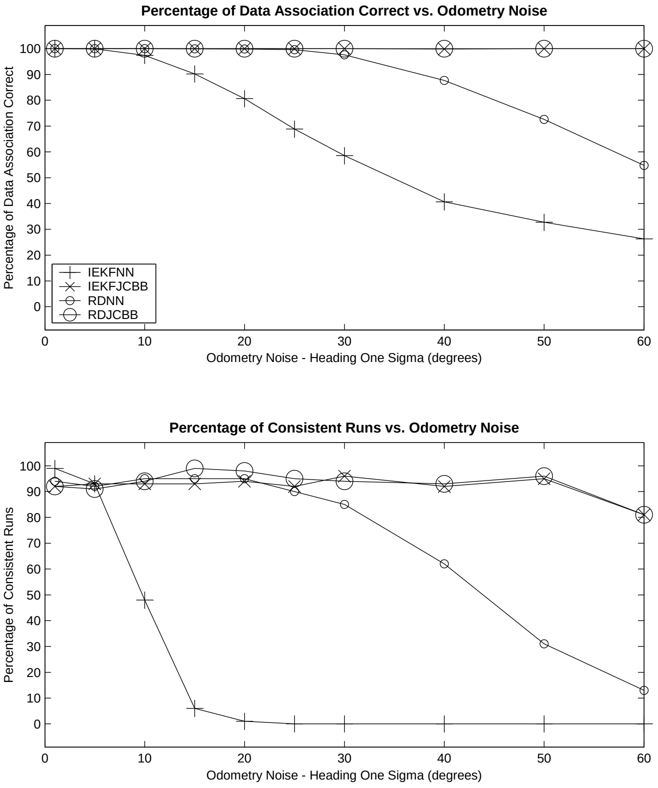

Fig. 5. Comparison for increasing odometry (rate-of-turn) noise.

Our initial simulation results show that: (i) our SLAMIDE algorithm significantly outperforms other methods which treat dynamic objects as clutter; (ii) our method for computing reversible data association remains consistent when other data association methods fail and (iii) our reversible data association provides excellent performance when clutter is present. Aside from simulation we have also successfully run our algorithms on a small set of real radar data with very promising initial results.

The first thing we would like to improve is the use of a Gaussian on average velocity for the likelihood model in our model selection. In practice this works well, however, a technique that does not introduce an extra threshold would be preferable, ideally it would work directly on the estimated distribution and the innovation sequence over the sliding window. Secondly, we have observed that the length of the sliding window in our experiments is often longer than it needs to be. We are currently considering an active threshold based on the convergence of parameters within the sliding window to select an appropriate length on the fly.

In summary, we have developed a method for robustly including dynamic objects directly into the SLAM estimate as opposed to treating them as outliers. The benefit of including dynamic objects in a single framework is clear for navigation and path planning; interestingly it also helps with localisation in highly dynamic environments, especially during short peri-

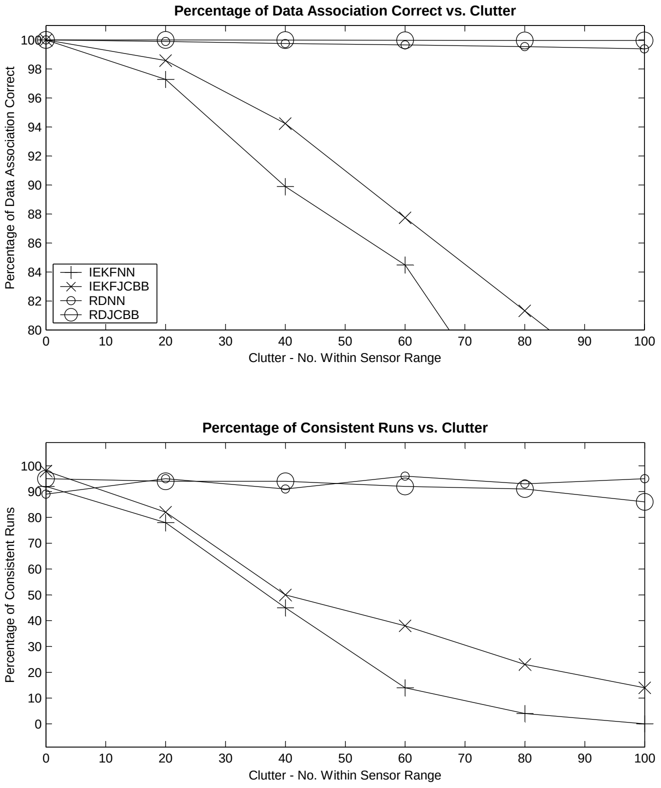

Fig. 6. Comparison for increasing clutter.

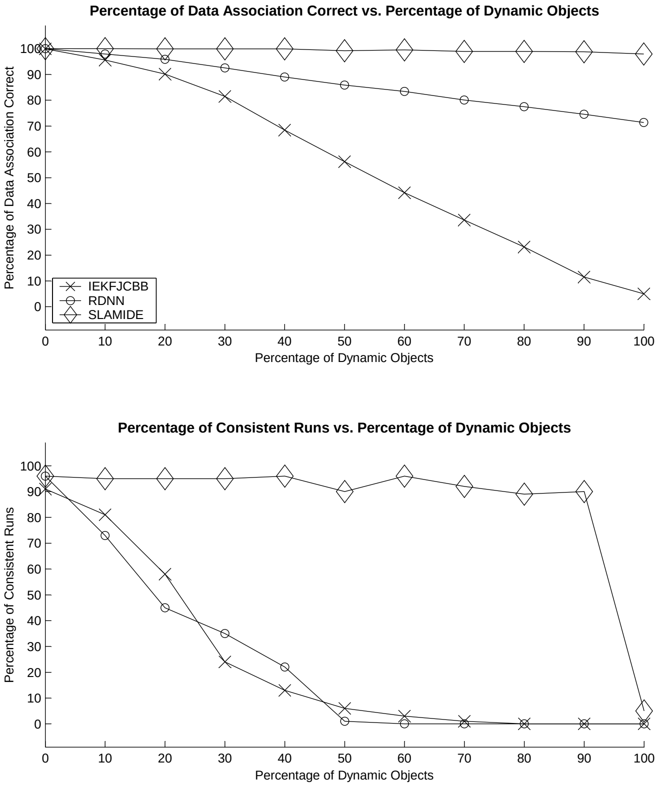

Fig. 7. Comparison for an increasing percentage of dynamic objects.

ods of time without any stationary landmark observations. Our longer term goal is to build a navigational aid using sliding window SLAM with reversible data association and reversible model selection to fuse data from a high performance pan tilt camera, marine radar, GPS and compass.

REFERENCES

[1] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[2] S. Thrun, W. Burgard and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[3] G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Comput. Stat. Data Anal.*, 14(3):315–332, 1992.

[4] T. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[5] F. Dellaert. The expectation maximization algorithm. Technical Report GIT-GVU-02-20, Georgia Institute of Technology, 2002.

[6] F. Dellaert and M. Kaess. Square root SAM. *International Journal of Robotics Research*, 2006.

[7] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints*. MIT Press, Cambridge MA, 1990.

[8] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

[9] Y. Bar-Shalom, T. Kirubarajan and X. R. Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[10] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Society*, 1977.

[11] J. Leonard and R. Rikoski. Incorporation of delayed decision making into stochastic mapping. In *Experimental Robotics VII, Lecture Notes in Control and Information Sciences*, 2001.

[12] M. Meila and D. Heckerman. An experimental comparison of several clustering and initialization methods. In *Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998.

[13] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.

[14] J. Neira and J. Tardos. Data association in stochastic mapping using the joint compatibility test. In *IEEE Trans. on Robotics and Automation*, 2001.

[15] W. S. Wijesoma, L. D. L. Perera and M. D. Adams. Toward multidimensional assignment data association in robot localization and mapping. In *IEEE Transactions on Robotics*, April 2006.

[16] L. L. Scharf and L. T. McWhorter. Geometry of the Cramer-Rao bound. *Signal Process.*, 31(3):301–311, 1993.

[17] G. Sibley, G. S. Sukhatme and Larry Matthies. Constant time sliding window filter SLAM as a basis for metric visual perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, April 2007.

[18] C. C. Wang, C. Thorpe and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, September 2003.

[19] R. Eustice, H. Singh, J. Leonard, M. Walter and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[20] D. Wolf and G. S. Sukhatme. Online simultaneous localization and mapping in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, Louisiana, April 2004.

This work is sponsored by Servowatch Systems Ltd.