IEEE Trans. Robotics and Automation Submission Cover Page

For Submission as a regular paper:

Title:
An Approximate Bayesian Method for Simultaneous Localisation and Mapping

Authors:
Somajyoti Majumder, Hugh Durrant-Whyte, Sebastian Thrun and Marc de Battista

Corresponding Author:
Hugh Durrant-Whyte
Australian Centre for Field Robotics (ACFR) J04,
The University of Sydney,
Sydney NSW 2006,
Australia.
Fax: +61-2-9351-7474
hugh@acfr.usyd.edu.au

Other Authors
Somajyoti Majumder and Marc de Battista are also with the ACFR at the address above.

Sebastain Thrun is with
Department of Computer Science,
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, USA,
thrun@heaven.learning.cs.cmu.edu

Abstract
This paper describes a Bayesian formulation of the Simultaneous Localisation and Mapping (SLAM) problem. Previously, the SLAM problem could only be solved in real time through the use of the Kalman Filter. This generally restricts the application of SLAM methods to domains with straight-forward (analytic) environment and sensor models. In this paper the Sum-of-Gaussian (SOG) method is used to approximate more general (arbitrary) probability distributions. This representation permits the generalizations made possible by Monte-Carlo methods, while inheriting the real-time computational advantages of the Kalman filter. The method is demonstrated by its application to sub-sea field data. The sub-sea data consists of both sonar and visual information of near-field landmarks. This is a particularly challenging problem incorporating diverse sensing modalities, amorphous environment features, and poorly known vehicle dynamics; none of which can be easily handled by Kalman filter-based SLAM algorithms.

An Approximate Bayesian Method for Simultaneous Localisation and Mapping

Somajyoti Majumder, Hugh Durrant-Whyte, Sebastian Thrun and Marc de Battista

Abstract— **This paper describes a Bayesian formulation of the Simultaneous Localisation and Mapping (SLAM) problem. Previously, the SLAM problem could only be solved in real time through the use of the Kalman Filter. This generally restricts the application of SLAM methods to domains with straight-forward (analytic) environment and sensor models. In this paper the Sum-of-Gaussian (SOG) method is used to approximate more general (arbitrary) probability distributions. This representation permits the generalizations made possible by Monte-Carlo methods, while inheriting the real-time computational advantages of the Kalman filter. The method is demonstrated by its application to sub-sea field data. The sub-sea data consists of both sonar and visual information of near-field landmarks. This is a particularly challenging problem incorporating diverse sensing modalities, amorphous environment features, and poorly known vehicle dynamics; none of which can be easily handled by Kalman filter-based SLAM algorithms.**

I. Introduction

Simultaneous Localisation and Mapping (SLAM) is a process by which a mobile platform can build a map of an environment and at the same time use this map to deduce it's location. In SLAM both the trajectory of the platform and the location of all landmarks are estimated on-line without the need for any *a priori* knowledge of location. SLAM is the subject of considerable current research activity [4], [5], [13].

The complexity of the SLAM estimation problem is potentially huge (of the dimension of the number of landmarks). Further, the structure of the SLAM problem is characterised by monotonically increasing correlations between landmark estimates. Thus the state space can not be trivially decoupled. For these reasons, there has been a significant drive to find computationally effective SLAM algorithms. This has been achieved through the development and use of the Kalman and extended Kalman filter as the estimation algorithms of choice in SLAM algorithms. In these developments, simplification in the time update step and locality in the observation update step have resulted in algorithms that can process thousands of land-marks in real time on PC level architectures [5], [6].

However, the Kalman filter approach comes with a number of limitations. Most notably, the inability to represent complex environment or feature models, the difficulty of faithfully describing highly skewed or multi-modal vehicle error models, and the inherent complexity of the resulting data association problem. A more general approach to vehicle navigation, which overcomes many of these limitations, is to consider navigation as a Bayesian estimation problem [19]. In this method, vehicle motion and feature observation are described directly in terms of the underlying probability density functions and Bayes theorem is used to fuse observation and motion information. These methods have demonstrated considerable success in some challenging environments [18].

Practically, Bayesian methods have generally been implemented using a combination of grid-based environment modelling and particle filtering techniques. While such approaches work well in low-dimensional problems such as localisation, they scale exponentially in the dimension of the state space. The underlying state space in SLAM is of dimension the number of land-marks and can not be trivially decoupled. Thus the application of particle filtering methods to SLAM is limited.

The work described in this paper was initially motivated by the session on navigation and subsequent discussions held at the 1999 International Symposium on Robotics Research (ISRR'99) [7]. The two alternative methods to navigation were each presented and following discussions highlighted both the advantages and disadvantages of each method as described in the preceding paragraphs. Of particular interest was an autonomous underwater vehicle (AUV) navigation example. In this case, the ability of the platform to undertake SLAM is essential as there is no *a priori* map, and no access to external reference (such as GPS) to provide position information for navigation. However, the AUV navigation problem is also characterised by very poor vehicle motion models, by highly unstructured and amorphous environments, and by poor quality sensor (sonar and visual) observations. Thus, the direct application of Kalman filter-based SLAM methods in sub-sea environments is difficult. Conversely, the substantial representational advantages of the Bayesian method in this application must be weighed against the insurmountable computational complexity of applying Bayesian estimation techniques with inefficient representations such as particles or grids to the sub-sea SLAM problem

The fundamental approach taken in this paper is to agree that the Bayesian methodology is indeed the correct representational form for the sub-sea SLAM problem but then to seek a *functional* representation for the underlying probability distributions which yields a computationally efficient SLAM algorithm. In this paper, the Sum of Gaussians (SOG) method of describing general probability distributions is employed. This is a well known method of dealing

Somajyoti Majumder, Hugh Durrant-Whyte, and Marc de Battista are with the Australian Centre for Field Robotics (ACFR) J04, The University of Sydney, Sydney NSW 2006, Australia, hugh@acfr.usyd.edu.au, Sebastain Thrun is with the Department of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA, thrun@heaven.learning.cs.cmu.edu

with non-Gaussian probability distributions in an efficient manner [1]. The SOG method has substantial computational advantages in allowing Kalman filter-based methods to be extended to more general probability distributions with richer representational abilities.

This paper begins by stating the full Bayesian formulation of the SLAM problem. Subsequently the SOG method is introduced and it is demonstrated that this can be used to effectively model AUV motion, typical sub-sea environments, and the sensor data returned by both sonar and vision data. The application of the SOG approximation to the full-Bayesian SLAM problem is then described. The results are explained through the use of field data obtained from an operational AUV. In conclusion, it is argued that the SOG method provides a computationally tractable implementation of the full Bayes SLAM or map building algorithm while maintaining all the representational advantages of such methods.

II. Bayesian Formulation of the SLAM problem

A Bayesian formulation of the SLAM problem is introduced. The structure of the problem is briefly described relative to conventional localisation or map building process.

A. Preliminaries

Consider a vehicle moving through an environment taking relative observations of a number of unknown landmarks using a sensor located on the vehicle. At a time instant k, the following quantities are defined:

- $\mathbf{x}_k$: The state vector describing the location and orientation of the vehicle.
- $\mathbf{u}_k$: The control vector, applied at time $k-1$ to drive the vehicle to a state $\mathbf{x}_k$ at time k.
- $\mathbf{m}_i$: A vector describing the location of the ith landmark whose true location is assumed time invariant.
- $\mathbf{z}_{ik}$: An observation taken from the vehicle of the location of the ith landmark at time k. When there are multiple landmark observations at any one time or when the specific landmark is not relevant to the discussion, the observation will be written simply as $\mathbf{z}_k$.

In addition, the following sets are also defined:

- The history of vehicle locations:

$$\mathbf{X}^k = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\} \ = \{\mathbf{X}^{k-1}, \mathbf{x}_k\} \tag{1}$$

- The history of control inputs:

$$\mathbf{U}^k = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} \ = \{\mathbf{U}^{k-1}, \mathbf{u}_k\} \tag{2}$$

- The set of all landmarks:

$$\mathbf{m} \ = \ \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_n\} \tag{3}$$

- The set of all landmark observations:

$$\mathbf{Z}^k = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} \ = \{\mathbf{Z}^{k-1}, \mathbf{z}_k\} \tag{4}$$

B. Definition of the SLAM problem

In probabilistic form, the Simultaneous Localisation and Map Building (SLAM) problem requires that the probability distribution

$$P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) \tag{5}$$

be computed for all times k. This probability distribution describes the joint posterior density of the landmark locations and vehicle state (at time k) given the recorded observations and control inputs up to and including time k together with the initial state of the vehicle. It is well known that that in the SLAM problem, the vehicle location and all of the landmark locations are highly correlated and so can not be considered independently [3], [12]. These correlations are fundamental to the structure and convergence of a SLAM algorithm.

In general, a recursive solution to the SLAM problem is desirable. Starting with an estimate for the distribution $P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1})$ at time $k-1$, the joint posterior, following a control $\mathbf{u}_k$ and observation $\mathbf{z}_k$, is to be computed using Bayes Theorem. This computation requires that a state transition model and an observation model are defined describing the effect of the control input and observation respectively.

The observation model describes the probability of making an observation $\mathbf{z}_k$ when the vehicle location and landmark locations are known, and is generally described in the form

$$P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}). \tag{6}$$

It is reasonable to assume that once the vehicle location and map are defined, observations are conditionally independent, and depend only on the map and the current vehicle state so that

$$\begin{aligned} P(\mathbf{Z}^k \mid \mathbf{X}^k, \mathbf{m}) &= \prod_{i=1}^{k} P(\mathbf{z}_i \mid \mathbf{X}^k, \mathbf{m}) \\ &= \prod_{i=1}^{k} P(\mathbf{z}_i \mid \mathbf{x}_i, \mathbf{m}). \end{aligned} \tag{7}$$

The motion model for the vehicle can be described in terms of a probability distribution on state transitions in the form

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \tag{8}$$

That is, the state transition may reasonably be assumed to be a Markov process in which the next state $\mathbf{x}_k$ depends only on the immediately proceeding state $\mathbf{x}_{k-1}$ and the applied control $\mathbf{u}_k$, and is independent of both the observations and the map. With these definitions and models, Bayes Theorem may be employed to define a recursive solution to Equation 5.

C. Formulation of the SLAM Problem

To derive a recursive update rule for the vehicle and map posterior, the chain rule of conditional probability is employed to expand the joint distribution of vehicle state, map

and observation in terms of the vehicle and map

$$\begin{aligned}
&P(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\
&= P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{z}_k, \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\
&= P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{z}_k \mid \mathbf{Z}^{k-1} \mathbf{U}^k, \mathbf{x}_0), \qquad (9)
\end{aligned}$$

and then in terms of the observation

$$\begin{aligned}
&P(\mathbf{x}_k, \mathbf{m}, \mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\
&= P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}, \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\
&= P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \qquad (10)
\end{aligned}$$

where the last equality employs the assumptions established for the sensor model in Equation 7.

Equating Equations 9 and 10 and rearranging gives

$$\begin{aligned}
&P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) \\
&= \frac{P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0)}{P(\mathbf{z}_k \mid \mathbf{Z}^{k-1}, \mathbf{U}^k)} \qquad (11)
\end{aligned}$$

The denominator in Equation 11 is independent of either the map or current vehicle state and can therefore be set to some normalising constant K. The total probability theorem can be used to rewrite the second term in the numerator in terms of the vehicle model and the joint posterior from time-step $k-1$ as

$$\begin{aligned}
&P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\
&= \int P(\mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1} \\
&= \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{m}, \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \\
&\times P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^k, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1} \\
&= \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \\
&\times P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1} \qquad (12)
\end{aligned}$$

where the last equality follows from the assumed independence of vehicle motion from map and observations, and from the causality of the vehicle control input on vehicle motion. Equation 12 is then substituted into Equation 11 to yield

$$\begin{aligned}
&P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) \\
&= K \cdot P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \\
&\times P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1}. \qquad (13)
\end{aligned}$$

Equation 13 provides a recursive expression for the calculation of the joint posterior $P(\mathbf{x}_j, \mathbf{m} \mid \mathbf{Z}^j, \mathbf{U}^j, \mathbf{x}_0)$ for the vehicle state $\mathbf{x}_j$ and map $\mathbf{m}$ at a time j based on all observations $\mathbf{Z}^j$ and all control inputs $\mathbf{U}^j$ up to and including time j. The recursion is a function of a vehicle model $P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k)$ and an observation model $P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})$. A related problem is to include the complete history of vehicle states [20] (a smoothed trajectory estimates) as $P(\mathbf{X}^k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k)$. However this is not required in the development below.

This process of time and observation update described in Equation Equation 13 is shown schematically in Figures 1 and 2. These figures show how the motion and observation models are captured and then employed in recursive computation of the posterior.

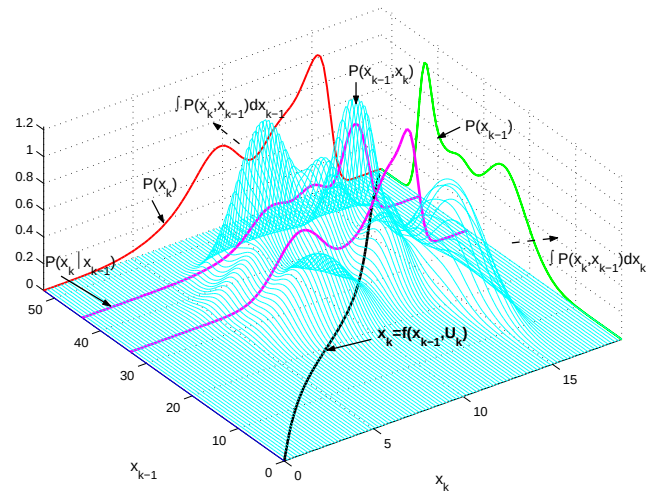


Fig. 1. *Time update step for the full Bayes filter. At a time $k-1$, knowledge of the state $\mathbf{x}_{k-1}$ is summarised in a probability distribution $P(\mathbf{x}_{k-1})$. A vehicle model, in the form of a conditional probability density $P(\mathbf{x}_k \mid \mathbf{x}_{k-1})$, then describes the stochastic transition of the vehicle from a state $\mathbf{x}_{k-1}$ at a time $k-1$ to a state $\mathbf{x}_k$ at a time k. Functionally, this state transition may be related to an underlying kinematic vehicle model in the form $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k)$. The figure shows two typical conditional probability distributions $P(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ on the state $\mathbf{x}_k(k)$ given fixed values of $\mathbf{x}_{k-1}$. The product of this conditional distribution with the marginal distribution $P(\mathbf{x}_{k-1})$, describing the prior likelihood of values of $\mathbf{x}_k$, gives the joint distribution $P(\mathbf{x}_k, \mathbf{x}_{k-1})$ shown as the surface in the figure. The total marginal density $P(\mathbf{x}_k)$ describes knowledge of $\mathbf{x}_k$ after state transition has occurred. The marginal density $P(\mathbf{x}_k)$ is obtained by integrating (projecting) the joint distribution $P(\mathbf{x}_k, \mathbf{x}_{k-1})$ over all $\mathbf{x}_{k-1}$. Equivalently, using the total probability theorem, the marginal density can be obtained by integrating (summing) all conditional densities $P(\mathbf{x}_k \mid \mathbf{x}_{k-1})$weighted by the prior probability $P(\mathbf{x}_{k-1})$ of each $\mathbf{x}_{k-1}$. The process can equally be run in reverse (a retroverse motion model) to obtain $P(\mathbf{x}_{k-1})$ from $P(\mathbf{x}_k)$ given a model $P(\mathbf{x}_{k-1} \mid \mathbf{x}_k)$.*

D. Map Building, Localisation and SLAM

Equation 5 should be compared to two related sub problems; map building and localisation.

The map building problem may be formulated as computing the probability distribution $P(\mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0)$. This assumes that the location of the vehicle $\mathbf{x}_k$ is known (or at least deterministic) at all times, subject to knowledge of initial location. A map $\mathbf{m}$ is then constructed by fusing observations from different locations. Conversely, the localisation problem may be formulated as computing the probability distribution $P(\mathbf{x}_k \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{m})$. This assumes that the landmark locations are known with certainty and the objective is to compute an estimate of vehicle location with respect to these landmarks.

The essential issue in the combined localisation and map-building problem is that solving the map building and lo-

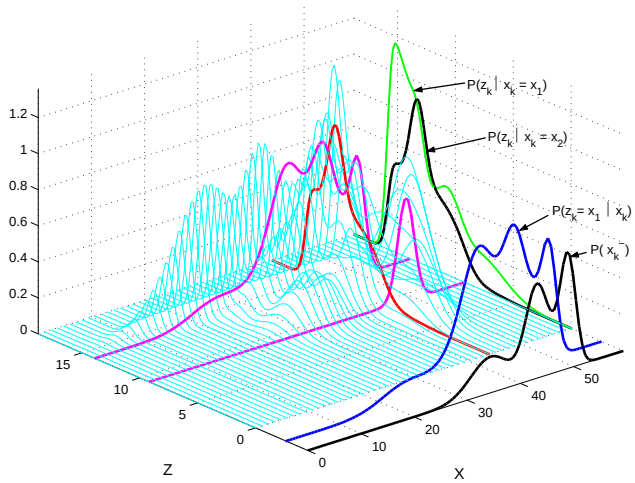


Fig. 2. *Observation update for the full Bayes filter. Prior to observation, an observation model in the form of the conditional density $P(\mathbf{z}_k \mid \mathbf{x}_k)$ is established. For a fixed value of $\mathbf{x}_k$, equal to $\mathbf{x}_1$ or $\mathbf{x}_2$ for example, a density function $P(\mathbf{z}_k \mid \mathbf{x}_k = \mathbf{x}_1)$ or $P(\mathbf{z}_k \mid \mathbf{x}_k = \mathbf{x}_2)$ is defined describing the likelihood of making the observation $\mathbf{z}_k$. Together the density $P(\mathbf{z}_k \mid \mathbf{x}_k)$ is then a function of both $\mathbf{z}_k$ and $\mathbf{x}_k$. This conditional density then defines the observation model. Now, in operation, a specific observation $\mathbf{z}_k = \mathbf{z}_1$ is made and the resulting distribution $P(\mathbf{z}_k = \mathbf{z}_1 \mid \mathbf{x}_k)$ defines a density function (now termed the likelihood function) on $\mathbf{x}_k$. This density is then multiplied by the prior density $P(\mathbf{x}_k^-)$ and normalised to obtain the posterior distribution $P(\mathbf{x}_k \mid \mathbf{z}_k)$ describing knowledge in the state after observation.*

calisation problems separately is not the same as solving the full SLAM problem

$$P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k) \neq P(\mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k) P(\mathbf{x}_k \mid \mathbf{Z}^k, \mathbf{U}^k). \quad (14)$$

The reason for this is that the vehicle location and map are not conditionally independent. This is clear from the nature of the observation model $P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})$ where the single observation $\mathbf{z}_k$ depends on both vehicle and landmark locations.

III. Solving the SLAM problem

There are a number of possible methods of solving Equation 13. Here we briefly describe two methods at the extremes of the solution space: The Extended Kalman Filter (EKF), and the Particle filter or Monte-Carlo method. Subsequently, the Sum-of-Gaussian (SOG) method is proposed as an appropriate compromise between these two methods.

A. The Extended Kalman Filter (EKF)

The basis for the EKF is to describe the vehicle motion model in terms of a kinematic model subject to zero mean uncorrelated (Gaussian) errors in the form

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \Longleftrightarrow \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \quad (15)$$

where $\mathbf{f}(\cdot)$ models vehicle kinematics and $\mathbf{w}_k$ motion disturbances, and to describe the observation model in terms of a geometric observation, again subject to zero mean uncorrelated (Gaussian) errors, in the form

$$P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) \Longleftrightarrow \mathbf{z}(k) = \mathbf{h}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k, \quad (16)$$

where $\mathbf{h}(\cdot)$ describes the geometry of the observation and $\mathbf{v}_k$ models observation error.

The key to the simplicity of the EKF solution is the fact that a product of Gaussian distributions is Gaussian, and the convolution of Gaussian distributions is also Gaussian. Thus, when the vehicle model is assumed Gaussian, the prediction stage (the integral or convolution in Equation 13) yields a Gaussian, and when the observation model is Gaussian, the update stage (the product in Equation 13) also yields a Gaussian. With these Gaussians described by their respective means and variances, the EKF proceeds to solve Equation 13 in terms of only an estimated mean and covariance

$$\begin{aligned} \begin{bmatrix} \hat{\mathbf{x}}_k \\ \hat{\mathbf{m}} \end{bmatrix} &= \mathrm{E}\begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix} \mid \mathbf{Z}^k, \\ \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xm} \\ \mathbf{P}_{xm}^T & \mathbf{P}_{mm} \end{bmatrix}_k &= \mathrm{E}\left[\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}} \end{pmatrix} \begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}} \end{pmatrix}^T \mid \mathbf{Z}^k \right] \end{aligned} \quad (17)$$

of the joint distribution $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0)$. While the computation of these means and variances is still significant for large numbers of landmarks, computationally efficient implementations have been developed which permit map building with many thousands of landmarks to be accomplished in real-time [6], [13].

The EKF however comes with many potential problems and limitations. The usual criticism that the non-linearity of vehicle motion and landmark observation models yield distributions that are not truly Gaussian is, in practice, rarely an issue. A more substantive problem is the difficulty of modelling natural environment features in a form that is amenable to use in an EKF. A second formidable problem is the fragility of the EKF method when faced with incorrect associations of observations to landmarks.

B. The Particle Filter Method

In contrast to the EKF, Particle filtering or Monte-Carlo methods aim to provide a complete representation of the joint posterior density using a large set of sample points, termed particles. These points provide a faithful approximation to the true shape of the full distributions employed. State propagation and observation models are also represented in the form of a sampled distribution. A number of related algorithms have been proposed to then fuse and propagate estimates in the general form of Equation 13. Of note is the sampling importance resampling (SIR) or boot strap filter algorithm for computation of the update stage in Particle filters [16]. A number of similar algorithms have been proposed; the Monte-Carlo filter [11] and the condensation algorithm [8] are good examples.

Of particular relevance to this paper is the work described in [20], [19], [18], where particle filtering algorithms have been developed and applied with great success to indoor mobile robot navigation problems. In the case of localisation, a grid-based environment model is constructed to represent probabilistic landmark information. The vehicle motion is also then described in probabilistic form. Successive observations are then used to compute a location posterior for the vehicle in terms of a set of samples or particles. Most importantly, this method overcomes a number of key problems with EKF localisation methods; data association (and the kidnapped robot problem) and severe non-linearities in vehicle motion.

Extending particle filter methods from localisation to map building is, however, complicated by the fact that the state space for the map is much larger than the state space for the vehicle alone. Sampling methods scale exponentially with state dimension and thus full posterior estimation, using particle filters for map building, is generally intractable.

A solution to this is to use the expectation-maximisation (EM) technique [2], [22]. This method computes a sequence of maps with successively increasing likelihood. It operates in two alternate steps. The expectation step ($E-step$) computes a joint likelihood function for the data and vehicle location conditioned on the current map and data. In the M-step, the most likely map is computed given the pose estimates from the E-step. The EM method is demonstrably robust [20]. However, the M-step in particular, which computes the most likely map over all possible locations, is computationally intensive and is not yet practical for real-time implementations.

C. The Sum of Gaussian (SOG) Method

One approach to reducing the computational complexity of Particle filter methods is to find a *functional* representation for probability distributions in the full Bayesian algorithm. That is, the probability distributions describing vehicle motion and sensor observations would be described as a series of functions approximating the underlying true density functions. The Sum of Gaussian (SOG) method provides such a representation. It is appropriate to note that the SOG method can approximate any distribution to an arbitrary (in probability) accuracy [15]. Further, SOG approximated Bayesian estimators achieve convergent optimality with respect to full Bayes estimators [1], [9].

A general Gaussian is described in the form

$$\begin{aligned} P(\mathbf{x}) &= \mathcal{G}(\mathbf{x};\overline{\mathbf{x}},\Sigma) \\ &= \frac{\exp\left(-\frac{1}{2}\left(\mathbf{x}-\overline{\mathbf{x}}\right)^T\mathbf{\Sigma}^{-1}\left(\mathbf{x}-\overline{\mathbf{x}}\right)\right)}{(2\pi)^{n/2}\mid\mathbf{\Sigma}\mid^{1/2}} \end{aligned} \tag{18}$$

where $\mathbf{x}$ is the state of interest, and $\overline{\mathbf{x}}$ and $\mathbf{\Sigma}$ are the mean and variance characterizing the distribution. A Sum of Gaussians (SOG) or Gaussian mixture is described in the form

$$\begin{aligned} P(\mathbf{x}) &= \sum_{i=1}^{n}\alpha_i\mathcal{G}_i(\mathbf{x};\overline{\mathbf{x}}_i,\mathbf{\Sigma}_i) \\ &= \sum_{i=1}^{n}\frac{\alpha_i\exp\left(-\frac{1}{2}\left(\mathbf{x}-\overline{\mathbf{x}}_i\right)^T\mathbf{\Sigma}_i^{-1}\left(\mathbf{x}-\overline{\mathbf{x}}_i\right)\right)}{(2\pi)^{n/2}\mid\mathbf{\Sigma}_i\mid^{1/2}} \end{aligned} \tag{19}$$

where α_i are a series of weights normally summing to 1. An example of a one-dimensional Gaussian sum is shown in Figure 3(a). The great advantage of using a SOG model is that the computations involved are straight-forward modifications of the standard EKF equations. This is clear from the properties of Gaussians and SOGs:
1. The product of two Gaussian distributions is Gaussian.
2. The product of two SOG distributions is a SOG distribution.
3. The convolution of two Gaussian distributions is Gaussian
4. The convolution of two SOG distributions is a SOG distribution.
1. is the well-known conjugate distribution property of a Gaussian. 2. Follows as multiplication distributes over addition. 3. Follows because the Fourier transform of a Gaussian is Gaussian and because convolution is multiplication in the frequency domain. 4. follows from the fact that convolution is a linear operator.

There are two significant computational issues with SOG models. First, the product (or convolution) of two mixtures with (say) N_1 and N_2 Gaussians, results in a new mixture with $N_1 \times N_2$ Gaussians. Thus, the number of Gaussians used to model the density increases at each iteration of the estimation algorithm (Equation 13). Consequently, this requires re-sampling of the distribution as shown in Figure 3(b). The second significant problem is that in a Gaussian mixture the component functions are not orthogonal to each other (the SOG model does not form a basis for the density function).

D. Implementation of the SOG Method

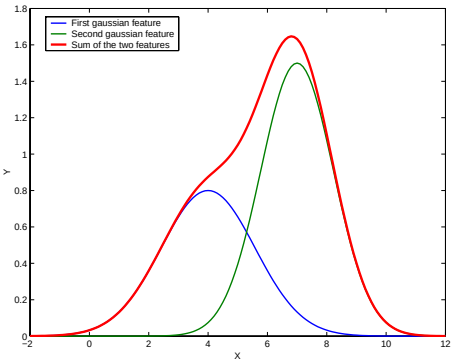
Practically, the SOG method is implemented by first defining a SOG model for sensor observations and for vehicle motion. The motion model is defined in the form

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1},\mathbf{u}_k) = \sum_{i=1}^{M_k}\alpha_i^v\mathcal{G}_i(\mathbf{x}_k;\overline{\mathbf{x}}_{k,i}^v,\mathbf{Q}_{k,i}) \tag{20}$$

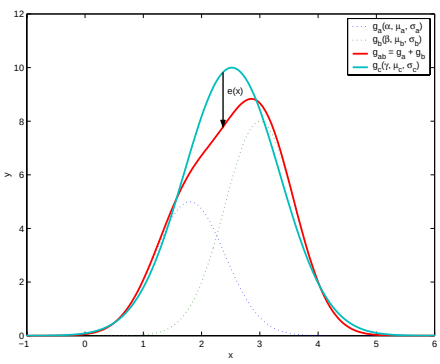
where the means $\overline{\mathbf{x}}_{k,i}^v = \overline{\mathbf{x}}_{k,i}^v(\mathbf{x}_{k-1},\mathbf{u}_k)$ and variances $\mathbf{Q}_{k,i}$=$\mathbf{Q}_{k,i}(\mathbf{u}_k)$ are functions of the previous state $\mathbf{x}_{k-1}$ and applied control input $\mathbf{u}_k$ computed from Equation 15 or from a small sample approximation such as the Distribution Approximation Filter [10].

The observation model is defined in the form

$$P(\mathbf{z}_k \mid \mathbf{x}_k,\mathbf{m}) = \sum_{i=1}^{N_k}\alpha_i^z\mathcal{G}_i(\mathbf{x}_k,\mathbf{m};\{\overline{\mathbf{x}}_{k,i}^z,\overline{\mathbf{m}}_{k,i}^z\},\mathbf{R}_{k,i}) \tag{21}$$



(a)



(b)

Fig. 3. Construction of SOG elements: (a) Example of a one-dimensional Gaussian Sum; (b) Example of re-sampling a SOG model to provide fewer sum elements. Generally, re-sampling aims to minimise some norm of the error $e(x)$ shown.

where the likelihood means $\overline{\mathbf{x}}^z_{k,i} = \overline{\mathbf{x}}^z_{k,i}(\mathbf{z}_k)$, $\overline{\mathbf{m}}^z_{k,i} = \overline{\mathbf{m}}^z_{k,i}(\mathbf{z}_k)$ and variances $\mathbf{R}_{k,i} = \mathbf{R}_{k,i}(\mathbf{z}_k)$ are functions of the observation made. As with the motion model, the observation model can be computed from either Equation 16 or from a small sample approximation.

The SOG algorithm now proceeds to recursively solve Equations 12 and 13 in the following manner. The joint posterior of vehicle and map at a time $k-1$ is assumed known in the form

$$\begin{aligned} &P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \\ &= \sum_{i=1}^{L_{k-1}} \alpha^x_i \mathcal{G}_i\{\mathbf{x}_{k-1}\mathbf{m}; \left[\overline{\mathbf{x}}^+_{k-1,i}, \overline{\mathbf{m}}^+_{k-1,i}\right] \\ &\left[\mathbf{P}^+_{k-1,i}, \mathbf{\Sigma}^+_{k-1,i}, \mathbf{S}^+_{k-1,i}\right]\} \end{aligned} \tag{22}$$

where

$$\mathbf{S}^+_{k-1,i} = E\left[\left(\mathbf{x}_{k-1} - \overline{\mathbf{x}}_{k-1,i}\right)\left(\mathbf{m} - \overline{\mathbf{m}}_{k-1,i}\right)^T \mid \mathbf{Z}^k\right] \tag{23}$$

is the cross-covariance between vehicle location and map estimate, and $\left[\overline{\mathbf{x}}^+_{k-1,i}, \overline{\mathbf{m}}^+_{k-1,i}\right]$ and $\left[\mathbf{P}^+_{k-1,i}, \mathbf{\Sigma}^+_{k-1,i}, \mathbf{S}^+_{k-1,i}\right]$ are the means and covariances of the joint distribution of map and vehicle. The $+$ superscript indicates that these are posterior (updated) values.

The time update step is computed by substituting Equation 20 and 22 into Equation 12 as

$$\begin{aligned} &P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \\ &= \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1} \\ &= \int \sum_{i=1}^{M_k} \alpha^v_i \mathcal{G}_i\{\mathbf{x}_k; \overline{\mathbf{x}}^v_{k,i}, \mathbf{Q}_{k,i}\} \sum_{j}^{L_{k-1}} \alpha^x_j \mathcal{G}_j\{\mathbf{x}_{k-1}, \mathbf{m}; \\ &\left[\overline{\mathbf{x}}^+_{k-1,j}, \overline{\mathbf{m}}^+_{k-1,j}\right], \left[\mathbf{P}^+_{k-1,j}, \mathbf{\Sigma}^+_{k-1,j}, \mathbf{S}^+_{k-1,j}\right]\} \mathrm{d}\mathbf{x}_{k-1} \\ &= \sum_{i=1}^{M_k} \sum_{j=1}^{L_{k-1}} \int \alpha^v_i \alpha^x_j \mathcal{G}_i\{\mathbf{x}_k; \overline{\mathbf{x}}^v_{k,i}, \mathbf{Q}_{k,i}\} \mathcal{G}_j\{\mathbf{x}_{k-1}, \mathbf{m}; \\ &\left[\overline{\mathbf{x}}^+_{k-1,j}, \overline{\mathbf{m}}^+_{k-1,j}\right], \left[\mathbf{P}^+_{k-1,j}, \mathbf{\Sigma}^+_{k-1,j}, \mathbf{S}^+_{k-1,j}\right]\} \mathrm{d}\mathbf{x}_{k-1} \\ &= \sum_{i=1,j=1}^{M_k, L_{k-1}} \alpha_{ij} \mathcal{G}_{ij}\{\mathbf{x}_k, \mathbf{m}; \{\overline{\mathbf{x}}^-_{k,ij}, \overline{\mathbf{m}}^-_{k,ij}\}, \mathbf{P}^-_{k,ij}\} \end{aligned} \tag{24}$$

The last step in Equation 24 is obtained by recognizing that the convolution of two Gaussian is a Gaussian with

$$\begin{aligned} \begin{bmatrix} \overline{\mathbf{x}}^-_{k,ij} \\ \overline{\mathbf{m}}^-_{k,ij} \end{bmatrix} &= \begin{bmatrix} \overline{\mathbf{x}}^v_{k,i} + \overline{\mathbf{x}}^+_{k-1,i} \\ \overline{\mathbf{m}}^+_{k-1,i} \end{bmatrix} \\ \mathbf{P}^-_{k,ij} &= \mathbf{Q}_{k,i} + \mathbf{P}^+_{k-1,j}. \end{aligned}$$

This is essentially identical to the prediction step of the Kalman filter for each mean and variance pairing. Note that the motion or state-transition model is captured in Equation 20 and does not form part of the update step.

The observation update step is computed by substituting Equation 21 and 24 in to Equation 13

$$\begin{aligned} &P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}^k, \mathbf{U}^k, \mathbf{x}_0) \\ &= KP(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \\ &\times P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}^{k-1}, \mathbf{U}^{k-1}, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1} \\ &= K \sum_{l=1}^{N_k} \alpha^z_l \mathcal{G}_l\{\mathbf{x}_k, \mathbf{m}; \left\{\overline{\mathbf{x}}^z_{k,l}, \overline{\mathbf{m}}^z_{k,l}\right\}, \mathbf{R}_{k,l}\} \\ &\times \sum_{i=1,j=1}^{M_k, L_{k-1}} \alpha_{ij} \mathcal{G}_{ij}\{\mathbf{x}_k, \mathbf{m}; \{\overline{\mathbf{x}}^-_{k,ij}, \overline{\mathbf{m}}^-_{k,ij}\}, \mathbf{P}^-_{k,ij}\} \\ &= K \sum_{l=1,i=1,j=1}^{N_k, M_k, L_{k-1}} \beta_{lij} \mathcal{G}_{lij}\{\mathbf{x}_k, \mathbf{m}; \left\{\overline{\mathbf{x}}^+_{k,lij}, \overline{\mathbf{m}}^+_{k,lij}\right\}, \mathbf{P}^+_{k,lij}\} \end{aligned} \tag{25}$$

where

$$\begin{aligned} \begin{bmatrix} \overline{\mathbf{x}}^+_{k,lij} \\ \overline{\mathbf{m}}^+_{k,lij} \end{bmatrix} &= \mathbf{P}^+_{k,lij} \begin{bmatrix} (\mathbf{R}_{k,l})^{-1} \overline{\mathbf{x}}^z_{k,l} + \left(\mathbf{P}^-_{k,ij}\right)^{-1} \overline{\mathbf{x}}^-_{k,ij} \\ (\mathbf{R}_{k,l})^{-1} \overline{\mathbf{m}}^z_{k,l} + \left(\mathbf{P}^-_{k,ij}\right)^{-1} \overline{\mathbf{m}}^-_{k,ij} \end{bmatrix} \\ \mathbf{P}^+_{k,lij} &= \left[\left(\mathbf{R}^-_{k,l}\right)^{-1} + (\mathbf{P}_{k,ij})^{-1}\right]^{-1} \end{aligned}$$

and β_{lij} are new mixture weights. This is again identical to the observation update step of the Kalman filter for each mean and variance pairing. Note that the observation model is captured in Equation 21 and does not, directly, form part of the update step. The new mixture weights α_{ij} and β_{lij} are computed as the product of component weights normalised by requiring each Gaussian in the mixture to integrate to 1 (as in Equation 18).

Equations 24 and 25 describe a recursive algorithm for computation of the full joint posterior as a sum of Gaussians. The initial conditions are obtained assuming initial vehicle location is captured in the distribution $P(\mathbf{x}_0)$, and the initial knowledge of the map is given by the first measurement set.

The SOG method first computes the time update (a convolution) for each combination of Gaussian in the state motion model and prior distributions. This then yields a further SOG mixture which is subsequently multiplied by the SOG observation distribution to yield a posterior distribution which is also a SOG mixture. the convolution and multiplication steps in the SOG method are directly equivalent to the time and observation update equations in the EKF, and indeed can be implemented simply as multiple instances of the conventional EKF algorithm. However, as is clear from the three-fold indices in Equation 25, the resulting posterior must then generally be re-sampled to produce a lower complexity mixture. Re-sampling is a weakness of this method as the SOG representation does not form an orthogonal basis, and consequently there is no unique re-sampling strategy. Practically, for the results presented here, a method of successive approximation and least-squares fitting is employed. This is described in more detail in [14].

A very important point to note is that the algorithmic flow is separate from the issue of sensor or platform modelling. The SOG method focuses on the algorithmic flow of convolution followed by multiplication described in Equation 25. This flow does not require any knowledge of how the models in Equations 22 and 24 are obtained. Indeed, these models could be obtained using any of Monte-Carlo, small sample or linearisation methods. Thus arguments about modelling approximations are not relevant to the applicability of SOG methods.

IV. Implementation of the SOG Method in Sub-Sea SLAM

AUV navigation is a particularly appropriate application of the SOG method. The underwater environment is very unstructured; landmark features consist of rocks and other objects on the sea-bed. Motion models for vehicles are poorly understood, and the quality of sub-sea sensing is generally very low. Together this makes it notably difficult to apply conventional EKF methods to the navigation problem. However, real-time SLAM is essential in AUV applications. There is generally no map of a sub-sea domain, and there are generally no widely available positioning systems such as GPS to locate a vehicle (leaving aside expensive and difficult to deploy short and long base-line sonar transponder networks). Thus deployment of a real-time sub-sea SLAM algorithm is one of the only methods of providing vehicle navigation and guidance information.

A. AUV, Sensors and Sensor Modeling

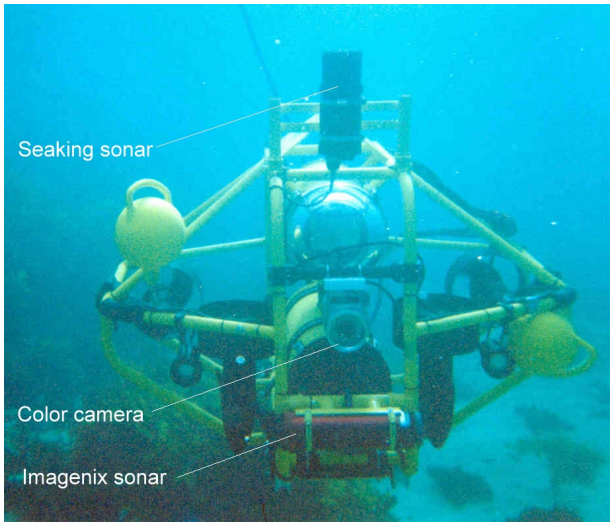
Oberon, the AUV employed in this implementation, is shown in Figure 4(a) [21]. Oberon is equipped with three main sensors a 585 kHz or 1210 kHz (user selectable) pencil-beam sonar, a 675 kHz fan beam sonar, and a specially constructed underwater color camera. The fan-beam simply provides altitude information. The pencil-beam sonar and the camera are co-registered to provide terrain data for the AUV navigation system (Figure 4(b)). Together the sensors provide terrain data in the form of range and bearing, as well as other properties such as texture and colour. The pencil-beam sonar has a beam-width of 1.8^0 and is mechanically scanned in azimuth at a maximum rate of 180 degrees/second. A typical scan, registered to the ground plane is shown in Figure 4(c). At each selected position, the sonar pings and a complete amplitude wave-form is returned. In this work, a Sum of Gaussian (SOG) model is fitted to the amplitude return of this ping to provide a model of the return observation. The ping and two SOG models of differing resolution are shown in Figure 4(d).

B. Processing and Analysis of Field Data

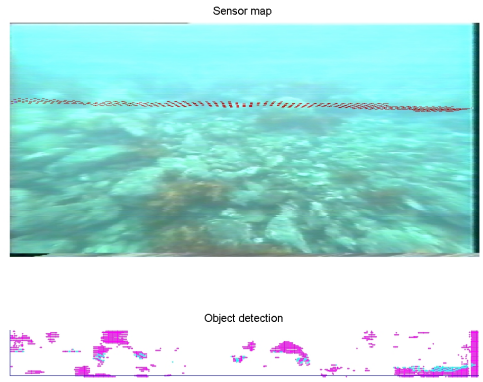
The operational environment for this work is a reef inlet on the Pacific shoreline near Sydney with a maximum depth of 5m. During the trials described here, the AUV followed a course parallel to the shoreline of the inlet, followed by a similar course in the opposite direction. An approximate vehicle path is shown in Figure 5(a). The length of each run is approximately 20-25metres. The experimental data consists of time stamped sonar data and co-located camera data, along with other navigation data including altitude, depth, yaw rate, compass heading and propeller control input.

The sensor model for the sonar data is a SOG in the form of Equation 21, an example of which is depicted in Figure 4(d). The motion model for the vehicle is taken to be a SOG derived from heading and thrust data in the form of Equation 20. This is a relatively weak model, but in this instant suffices at the typical speed and update rate of the AUV. After each sonar scan (approximately every two seconds), a local SOG terrain map is obtained from data such as that shown in Figure 4(c). This map data is propagated through Equation 13 using the AUV motion model. When new data is obtained, it is first converted into SOG form and multiplied by the propagated data to yield a posterior SOG. This SOG is then re-sampled before the next iteration of the algorithm.

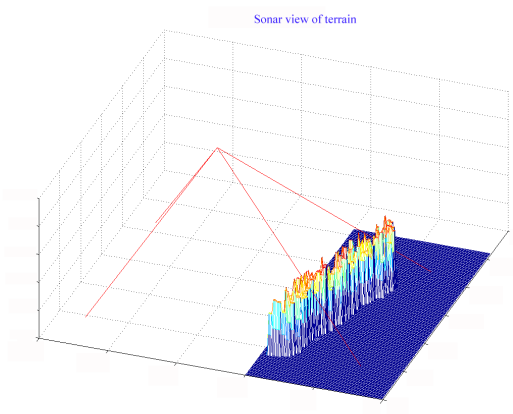
Figure 5(a) shows the state of the SOG terrain model after the first 100 seconds of motion. The sides of the inlet in the neighbourhood of the vehicle are clearly visible.



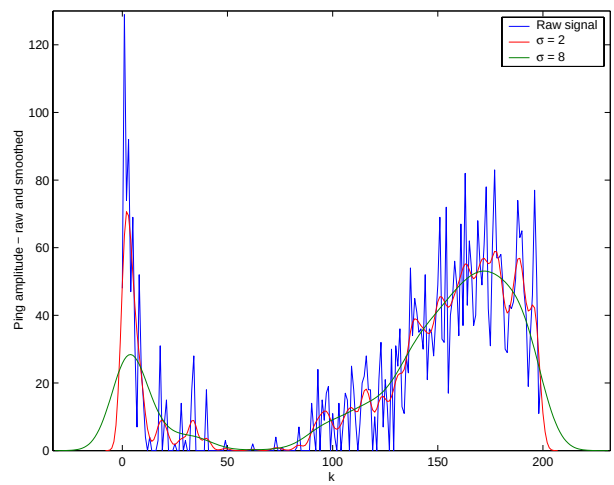
(a)



(b)



(c)



(d)

Fig. 4. Processing of experimental field data: (a) The experimental AUV, Oberon, showing location of the two sonars and vision system; (b) A typical data set showing sonar range data registered on camera image; (c) Projection of sonar data on to the ground plane, used for providing landmark data; (d) Fitting of Sum of Gaussian (SOG) distribution to a sonar ping (two possible resolutions shown)

Also coming into view is the first of a set of man-made dihedrals. These are placed in the environment to provide a measure of ground-truth for the SLAM algorithm. Figure 5(b) shows the SOG model after 300 seconds of mission duration. Figure 5(c) shows the state of the map at the furthest excursion of the vehicle along the reef structure. The "SOG" nature of the terrain model is evident both in the surrounding reef structure and in the strongly reflective dihedral near the centre of the terrain image. Figure 5(d) shows the resulting SOG terrain map at the end of the mission once the vehicle has returned to near it's starting position. The reconstructed inlet structure should be compared to the overlay of points shown in Figure 6(a). The string of dihedrals positioned along the mission trajectory stand out as highly peaked Gaussians (these too can be compared to Figure 5(a)). This highlights the fact that for well-behaved point features, the SOG method reduces to a conventional EKF with point landmarks. There are approximately 90 Gaussians in the re-sampled terrain. Re-sampling could be made both more or less stringent depending on the required application. The computational complexity involved is therefore equivalent to an EKF SLAM formulation with 90 landmarks.

V. Discussion and Conclusion

The terrain maps constructed using the SOG method appear to provide a quite general and robust representation of unstructured environments.

At the heart of the proposed method is the idea that the the posterior density is most efficiently represented as

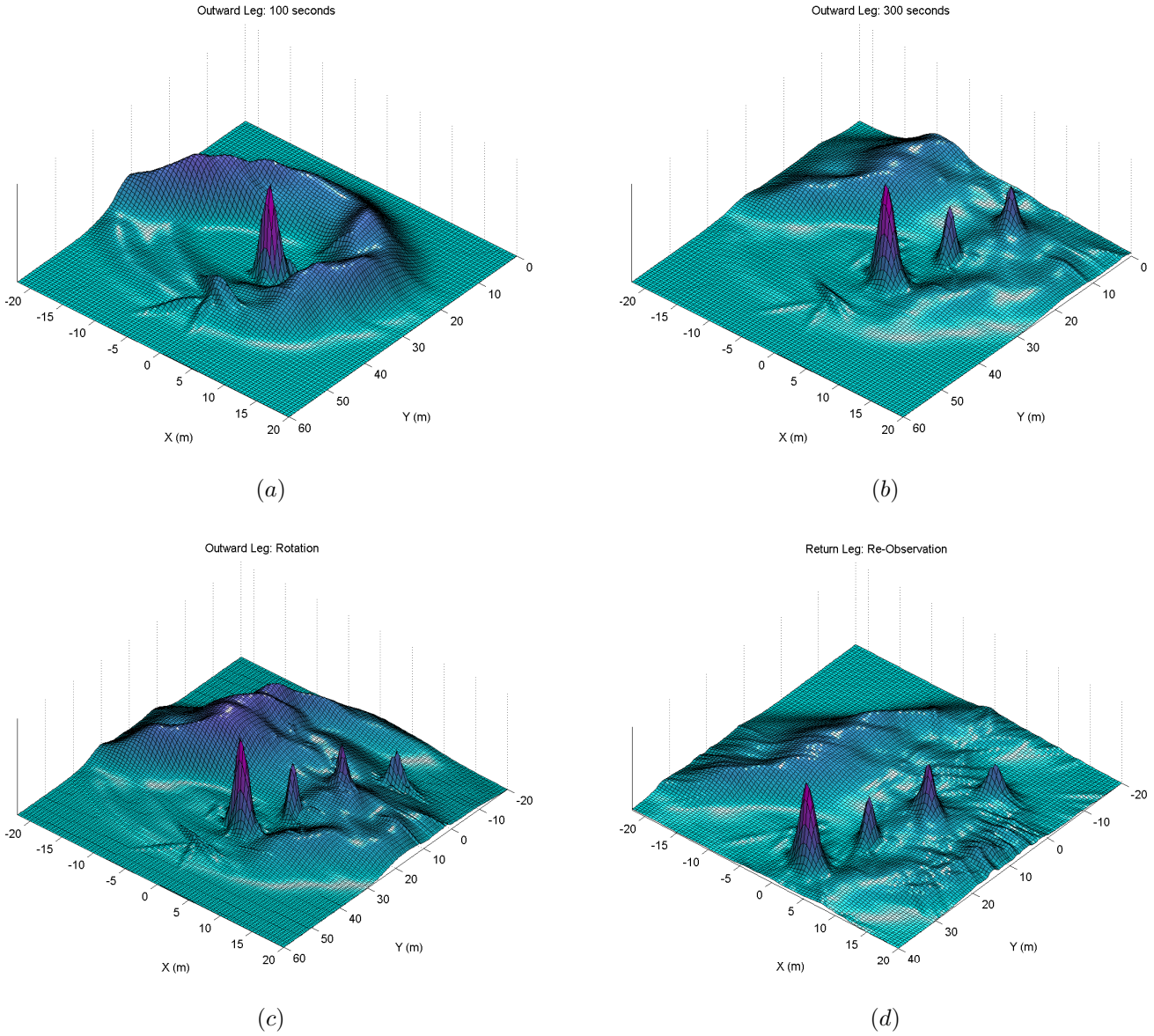


Fig. 5. Construction of a SOG terrain map from field trial: (a) Constructed SOG terrain map after the first 100s of motion; (b) Constructed SOG terrain map after 300s of motion;(c) Constructed SOG terrain map after at point of maximum excursion for the vehicle; (d) Constructed SOG terrain map after return and completion of mission. Note that the large peaks in this image correspond to the large returns in the registered set (Figure 6(a) below). These are due to the placement of man-made dihedrals at these locations to allow a ground truth path to be estimated.

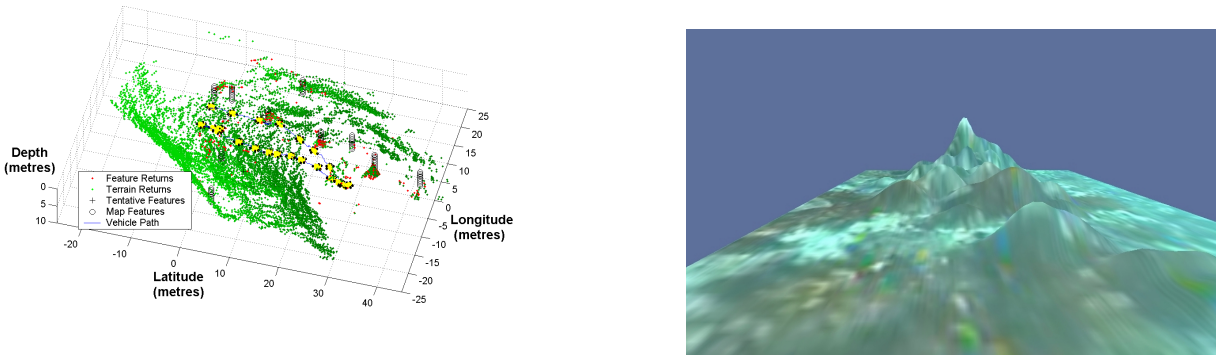
a function, rather than as a set of samples or indeed a grid. Representation as a function allows for a compact representation and efficient computation of prediction and update stages. The advantage of describing this function as a sum of Gaussians is that the computation simply reduces to multiple instances of a conventional Kalman filter. However, sums of Gaussians do not form an orthogonal series and so re-sampling to change resolutions is not always straight-forward. Other representations may be more appropriate.

The general Bayesian formulation of the SLAM problem goes some way to addressing the data association problem. Note that:

1. Features identified in successive scans combine to form an overall, multi modal, distribution which describe the main spatial characteristic of the map.
2. Spatially isolated features, which are observed only occasionally, are not reinforced and remain isolated.
3. A feature which is significant in one scan but which does not continue over successive scans, decreases in importance and its contribution to the map is of low significance.
4. Feature maps can be constructed in real-time and can be stored in a very compact form.

The problem of assigning measurements to landmarks is usually complex and is often the main source of fragility in a navigation algorithm. In contrast, the Bayesian (and specifically SOG) method deals with data association as simply a process of reinforcement or inhibition of density functions [14].

There are a large number of questions and issues raised



(a) (b)

Fig. 6. (a) Data set shown in the form of registered maximum returns together with a best computed vehicle path (Image courtesy Stefan Williams); (a) SOG estimate of AUV trajectory; (b) visual texture registered on SOG terrain image.

by the implementation of a full Bayes method in SLAM. Most importantly, is whether the full Bayes algorithm inherits the convergence properties of the Kalman filter-based SLAM methods [4]. This would require the terrain map to converge in probability monotonically and the platform location error to be bounded. A second issue is the incorporation of non-spatial feature descriptions as part of the map-building problem. A probability distribution provides a natural description of complex environment properties including texture, colour or reflectivity. It seems natural to extend a density function to higher dimensions than simply landmark location. Figure 6(b) shows an example of such a terrain map with texture data derived from vision overlayed on the SOG terrain map. Another element of the general density modeling is to use external information, such as reports or constraints to provide additional 'virtual' sensor data [17].

References

[1] D.L. Alspace and H.W. Sorenson. Gaussian filters for nonlinear filtering problems. *IEEE Trans. Automatic Control*, 17(2):439–448, 1972.

[2] R.A. Boyles. On the convergenece of the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 45(1):47–50, 1983.

[3] G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csobra. An experimental and theoretical investigation into simultaneous localisation and map building (SLAM). In P. Corke and J. Trevelyan, editors, *Experimental Robotics VI (ISER'99)*, pages 265–274. Springer-Verlag, 2000.

[4] G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csobra. A solution to the simultaneous localisation and mapping (SLAM) problem. *IEEE Trans. Robotics and Automation*, 17(3):229–241, 2001.

[5] J. Guivant, E.M. Nebot, and S. Baiker. Localization and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583, 2000.

[6] 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*, 17(3):242–257, 2001.

[7] J. Hollerbach and D. Koditscheck (Editors). *Robotics Research, The Ninth International Symposium (ISRR'99)*. Springer-Verlag, 2000.

[8] M. Isard and A. Blake. Condensation–conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[9] K Ito and K. Xiong. Gaussian filters for non-linear filtering problems. *IEEE Trans. Automatic Control*, 45(5):910–927, May 2000.

[10] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte. A new method for the non-linear transformations of means and covariances in filters and estimators. *IEEE Trans. Automatic Control*, 45(3):477–482, March 2000.

[11] G. Kitagawa. A Monte-Carlo filter and smoother for non-Gaussian non-linear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[12] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localisation for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS)*, pages 1442–1447, Osaka, Japan, 1991.

[13] J.J. Leonard and H.J.S. Feder. A computational efficient method for large-scale concurrent mapping and localisation. In J. Hollerbach and D. Koditscheck, editors, *Robotics Research, The Ninth International Symposium (ISRR'99)*, pages 169–176. Springer-Verlag, 2000.

[14] S. Majumder. *Sensor Fusion and Feature based Navigation for Subsea Robots*. PhD thesis, Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney, Australia, 2001.

[15] V. Mazya and G. Schmidt. On approximate approximations using Gaussian kernels. *IMA J. Numerical Analysis*, 16(1):13–29, 1996.

[16] D. Salmond, N. Gordon, and A.F.M. Smith. A novel approach to non-linear non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(1):107–113, 1993.

[17] L.D. Stone, C.A. Barlow, and T.L. Corwin. *Bayesian Multiple Target Tracking*. Artech House, 1999.

[18] S. Thrun. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, November 2000.

[19] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4), 2000.

[20] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1):29–53, 1998.

[21] S.B. Williams, P. Newman, G.Dissanayake, and H.F. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1793–1798, San Francisco, USA, April 2000.

[22] C.F.J. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95–103, 1983.