# Thesis Project

**Provisional Thesis Title:**
Simultaneous Localization and Mapping
for Mobile Robots in Urban Settings.

**Student:** Rafael Valencia Carreño
**Thesis Advisor:** Juan Andrade Cetto
**PhD Program:** Control, Visió i Robòtica
**Department:** Enginyeria de Sistemes, Automàtica i Informàtica Industrial (ESAII), Institut de Robòtica i Informàtica industrial (IRI)
**University:** Universitat Politècnica de Catalunya (UPC)

**September 5, 2008**

Institut de Robòtica i Informàtica Industrial

## Abstract

A truly autonomous robot should be capable of building a consistent map of the outside world while simultaneously determining its location within this map. However, to achieve this task the robot requires to select the appropriate motion commands during navigation in order to maximize its knowledge about the external world while at the same time reducing its own localization uncertainty. The objective of this thesis is to contribute to the state of the art of the SLAM problem with a novel approach to perform 6 degrees-of-freedom autonomous exploration in large outdoor urban environments, taking into account the problem of data association, the study of strategies to tackle scalability, the computational complexity of stochastic state estimation with large state spaces, the use of entropy-based exploration strategies, and the nonlinearities in perception and action models



**Institut de Robòtica i Informàtica Industrial (IRI)**
Consejo Superior de Investigaciones Científicas (CSIC)
Universitat Politècnica de Catalunya (UPC)
Llorens i Artigas 4-6, 08028, Barcelona
Spain

Tel (fax): +34 93 401 5750 (5751)
`http://www.iri.upc.edu`

**Corresponding author:**

Rafael Valencia Carreño
tel: +34 93 4015780
`rvalencia@iri.upc.edu`
`http://www.iri.upc.edu/people/rvalencia`

# Contents

# 1 Introduction

An autonomous robot should be able of representing and reasoning about its outside world and determining its location within this environment. Although very accurate sensors, the use of fixtures, and calibration points may help, they are still error-prone and usually costly. An alternative approach is to use multiple, overlapping, lower resolution sensors and to combine the spatial information (including the uncertainty) from all sources to obtain the best possible spatial estimate. The Simultaneous Localization and Mapping (SLAM) problem seeks to solve this issue.

SLAM is the problem of building a map of an unknown environment by a robot while at the same time being localized relative to this map. Thus the main advantage of SLAM is that it eliminates the need for artificial infrastructure or a priori topological knowledge of the environment. Autonomy however, also requires the robot to select the appropriate robot commands during navigation so as to maximize its knowledge about the external world while at the same time reducing its own localization uncertainty.

Solving the SLAM problem in conjunction with the selection of the appropriate motion commands by the robot itself, in order to jointly optimize for map coverage and active localization, can bring the following possibility closer: placing a robot at an unknown location in an unknown environment and then have it build a map, using only relative observations of the environment, and then to use this map simultaneously to navigate. In addition, a solution to this problem would be of inestimable value in a range of applications where absolute position or precise map information is unobtainable, including, amongst others, driving hundreds of kilometers under dense forest, mapping a whole city, including underpasses and tunnels, sometimes without recourse to global positioning systems (GPS), autonomous planetary exploration, subsea autonomous vehicles, and autonomous all-terrain vehicles in tasks such as mining and construction.

The main research topic of this thesis is autonomous exploration for outdoor vision-based 6 degrees-of-freedom simultaneous localization and mapping, using mainly visual information, taking into account the difficult problem of data association for large outdoor scenes, the study of strategies based on information theory metrics to tackle scalability by marginalizing robot poses from the map, the computational complexity of stochastic state estimation with large state spaces, the use of entropy-based exploration strategies, and the inherent nonlinearities in perception and action models.

This proposal is organized as follows. Section 2 presents the goal to be reached in the thesis. Section 3 is devoted to a literature survey on the state of the art in SLAM, considering the estimation process, Active SLAM solutions, and the environment representation. Section 4 states our approach to solve some of the identified open issues in the SLAM problem. In Section 5, we show a state-of-the-art implementation, which represent part of our previous work, and serves as a backbone starting point for state estimation in a trajectory-oriented SLAM. In Section 6 we present a calendar of the necessary tasks to achieve the objective of this thesis.

# 2 Objective

The main objective of this thesis is to contribute to the state of the art of the SLAM problem with a novel method to perform autonomous exploration for outdoor vision-based 6 degrees-of-freedom Simultaneous Localization and Mapping. Our approach will consist of a trajectory-oriented SLAM, whose main observations will be vision-based relative pose constraints. The proposed solution will take into account the following:

- Data association to close loops in large urban scenes by using information theory metrics.
- A robot pose marginalization strategy towards a long-term SLAM algorithm.
- Optimal exploration techniques for SLAM.
- The nonlinearities of perception and action models

The resulting algorithms will be evaluated both in simulations and experimentally with a robotic platform from the Institut de Robòtica i Informàtica industrial (IRI).

This thesis is part of the URUS (Ubiquitous Robotics in Urban Settings) [108] project at IRI, funded by the EU, and coordinated by Prof. Alberto Sanfeliu. Our contribution in URUS will be in the form of new robust mapping algorithms for outdoor mobile robots in pedestrian areas, using computer vision and stochastic state estimation tools.

# 3 State of the Art: Literature Survey

The present section is organized as follows: in Subsection 2.1 we state the SLAM problem and identify the challenges that it imposes. In Subsection 2.2 we overview the most popular existing solutions for the SLAM problem, considering estimation, exploration issues, and environment representation. Finally, in Subsection 2.3 we discuss the main conclussions of the literature survey and point out open research problems.

## 3.1 Problem Definition

Simultaneous Localization and Mapping (SLAM) is the problem of building a map of an unknown environment by a robot while at the same time being localized relative to this map. Although this problem is commonly abbreviated as SLAM, it was initially, during the second half of the 90's, also known as "Concurrent Mapping and Localization", or CML. The SLAM problem arises when the robot does not have access to a map of the environment, nor does it know its own pose. Instead, all it is given are measurements and controls.

The SLAM problem is characterized by uncertainty and sensor noise, therefore, virtually all state-of-the-art algorithms for SLAM are probabilistic. They employ probabilistic models of the robot and its environment, and they rely on probabilistic inference for turning sensor measurements into maps. Some authors make the probabilistic thinking very explicit, others use techniques that on the surface do not look specifically probabilistic.

The dominant scheme used in SLAM is the Bayes filter. The Bayes filter extends Bayes rule to temporal estimation problems. It is a recursive estimator for computing a sequence of posterior probability distributions over quantities that cannot be observed directly, such as a map.

Assuming a static world, the classical formulation for the SLAM problem requires that the probability distribution

$$p(\mathbf{x}_k, \mathbf{m} | Z^k, U^k, \mathbf{x}_0), \tag{1}$$

be computed for all times $k$. This probability distribution describes the joint posterior density of vehicle position $\mathbf{x}_k$ at time $k$ and the landmark locations $\mathbf{m}$ (i.e. the map) given the history of observations $Z^k$ until time $k$, the history of motion commands $U^k$ up to time $k$, and the initial pose $\mathbf{x}_0$.

To solve the SLAM problem, the robot needs to be endowed with models that describe the effect of the control input and the observations, that is, a state transition model and an observation model, respectively.

The observation model describes the probability of making an observation $\mathbf{z}_k$ when the vehicle location and landmark locations are known. It is assumed that, once the vehicle location and map are defined, observations are conditionally independent given the map and the current vehicle state. The observation model is generally described in the form

$$p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}). \tag{2}$$

The motion model for the vehicle is described in terms of a state transition probability distribution in the form

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k). \tag{3}$$

That is, the state transition is assumed to be a Markov process in which the next state $\mathbf{x}_k$ depends only on the immediately preceding state $\mathbf{x}_{k-1}$ and the applied control $\mathbf{u}_k$, and is independent of both the observations and the map.

Following the Bayes Filter framework, Equation 1 is calculated recursively in two steps, as follows:

**Prediction**

$$p(\mathbf{x}_k, \mathbf{m}|Z^{k-1}, U^k, \mathbf{x}_0) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)\, p(\mathbf{x}_{k-1}, \mathbf{m}|Z^{k-1}, U^{k-1}, \mathbf{x}_0)\, d\mathbf{x}_{k-1}. \tag{4}$$

**Measurement Update**

$$p(\mathbf{x}_k, \mathbf{m}|Z^k, U^k, \mathbf{x}_0) = \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m})\, p(\mathbf{x}_k, \mathbf{m}|Z^{k-1}, U^k, \mathbf{x}_0)}{p(\mathbf{z}_k|Z^{k-1}, U^k, \mathbf{x}_0)}. \tag{5}$$

The first step calculates the prior probability distribution $p(\mathbf{x}_k, \mathbf{m}|Z^{k-1}, U^k, \mathbf{x}_0)$ based on the posterior probability $p(\mathbf{x}_{k-1}, \mathbf{m}|Z^{k-1}, U^{k-1}, \mathbf{x}_0)$ computed one step-time before and the state transition probability distribution, using the Chapman-Kolmogorov equation. This step is called Prediction, Control Update, or Time-update.

The second step is called Measurement Update and it simply employs Bayes' Theorem, based on the observation model distribution and the prior distribution $p(\mathbf{x}_k, \mathbf{m}|Z^{k-1}, U^k, \mathbf{x}_0)$, to compute the joint posterior distribution $p(\mathbf{x}_k, \mathbf{m}|Z^k, U^k, \mathbf{x}_0)$, considering the assumption of Markovity for the state $\mathbf{x}_k$.

The above formulation is also known as the online SLAM problem. Other formulations of the SLAM problem have been defined in the literature, such as full SLAM problem or trajectory-oriented SLAM. In the full SLAM problem instead of estimating the present robot location, the entire robot path, together with all landmaks locations, are estimated. The trajectory-oriented SLAM is a delayed-state SLAM formulation, that is, we only estimate the entire robot path, and use observations to landmarks only to refine the path estimates. This formulations is usually employed when the size of features is much bigger than the number of robot poses. Furthermore, the former approaches are passive SLAM formulations, in that the robot is only externally commanded. On the other hand, an active SLAM formulation would require for the robot itself to select the control commands that reduce the uncertaintty about its environment and its own pose. Other refinements to the SLAM problem may take into account a dynamic environment or multi-robot mapping.

The SLAM problem is considered one of the hardest perceptual problems in robotics [121]. A key challenge in SLAM arises from the nature of the measurement noise. The noise in different measurements is not statistically independent. If this were the case, a robot could simply take more and more measurements to cancel out the effects of the noise. This problem occurs because errors in control accumulate over time, and they affect the way future sensor measurements are interpreted; therefore, whatever a robot infers about its environment is plagued by correlated errors. Accommodating such systematic errors is key to building maps successfully, and it is also a key complicating factor in robotic mapping.

The second complicating aspect of SLAM arises from the high dimensionality of the entities that are being mapped. If a the map is built upon the description of major topological entities, such as corridors, intersections, rooms and doors, a few dozen entities might suffice. A detailed two-dimensional (2D) floor plan, which is an equally common representation of robotic maps, often requires thousands of entities. But a detailed three-dimensional (3D) visual map of a building may easily require millions of entities. From a statistical point of view, each such entity adds a few dimensions to the underlying estimation problem. Thus, the mapping problem can be extremely high-dimensional.

Another challenge in SLAM is the correspondence problem, also known as the data association problem. The correspondence problem is the problem of determining if sensor measurements taken at different points in time correspond to the same physical object in the world. If a robot robot attempts to map a large cyclic environment, when closing the loop, the robot has to find out where it is relative to its previously built map. This problem is complicated by the fact that at the time of cycle closing, the robot's accumulated pose error might be very large. The correspondence problem is difficult, since the number of possible hypotheses can grow exponentially over time.

Changing environments is another challenge. Some changes may be relatively slow, such as the change of appearance of a tree across different seasons, or the structural changes that most office buildings are subjected to over time. Others are faster, such as the change of door status or the location of furniture items, such as chairs. Even faster may be the change of location of other agents in the environment, such as cars or people. The dynamism of robot environments creates a big challenge, since it adds yet another way in which seemingly inconsistent sensor measurements can be explained. The predominant SLAM paradigm relies on a static world assumption, in which the robot is the only time-variant quantity (and everything else that moves is just noise).

Another problem arises from the fact that robots must choose their way during mapping. The task of generating robot motion in the pursuit of maximizing robot's knowledge about the external world is called robotic exploration. Exploration for SLAM is commonly referred to as active SLAM or integrated exploration. Although optimal robot motion is relatively well-understood in fully modeled environments, exploring robots have to cope with partial and incomplete models. Hence, any viable exploration strategy has to be able to accommodate contingencies and surprises that might arise during map acquisition. For this reason, active SLAM is a challenging problem. When choosing where to move, various quantities have to be traded off: the expected gain in map information, the time and energy it takes to gain this information, the possible loss of pose information along the way, and so on.

## 3.2 Approaches to SLAM

Solutions to the SLAM problem can be distinguished along many different dimensions. Here we present a literature survey of solutions taking into account the estimation process, where we identify three main paradigms [124]: Kalman Filter, Particle Filters, and Graph-Based SLAM. We also address solutions to Active SLAM and solutions classified by the way they represent the environment.

### 3.2.1 Kalman Filter

A classic solution to the SLAM problem is based on Kalman Filters, which are Bayes Filters whose distributions are Gaussians. This approach can be traced back to a series of works by Smith *et al.* [115, 116] and Mourtalier and Chatila [93]. In the following years, a number of researchers developed this approach further [16, 17, 32, 33, 35, 78, 96].

Kalman Filter relies on three basic assumptions to guarantee that every posterior probability is always a Gaussian. First, the motion model must be linear with added Gaussian noise, and the same characteristics must also apply to the observation model. Second, the initial uncertainty must be Gaussian (i.e. Gaussian assumption). Third, the current state is conditionally independent of all earlier states given the immediately previous state (i.e. Markov assumption).

Usually, the robot pose is governed by a nonlinear trigonometric function that depends on the previous pose and the control command. Similarly, sensor measurements in robotics are usually nonlinear, with non-Gaussian noise. To accommodate such nonlinearities, Kalman filters approximate the robot motion and sensor models through a first degree Taylor series expansion. The resulting Kalman Filter is known as Extended Kalman Filter and the solution to SLAM using this filter is known as EKF-SLAM.

The EKF-SLAM method will yield the following distribution

$$p(\mathbf{y}_k|Z^k, U^k, \mathbf{x}_0) = \mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}_k, \mathbf{P}_k), \tag{6}$$

where the state vector $\mathbf{y}_k$ contains the robot pose $\mathbf{x}_k$ at time step $k$ and a vector $\mathbf{m}_k$ of map features up to time $k$, that is,

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{m}_k \end{bmatrix}. \tag{7}$$

Thus, Equation 1 is represented as a Gaussian distribution with mean $\hat{\mathbf{y}}_k$ and covariance $\mathbf{P}_k$.

The basis for the EKF-SLAM method is to describe the vehicle motion in the form

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) \longleftrightarrow \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \tag{8}$$

where $f$ models vehicle kinematics and where $\mathbf{w}_k$ is zero mean white noise, with covariance $\mathbf{Q}_k$, used to accomodate for higher order terms and modelling errors.

The observation model is described in the form

$$p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m}) \longleftrightarrow \mathbf{z}_k = h(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k, \tag{9}$$

where $h$ describes the geometry of the observation and where $\mathbf{v}_k$ are additive, zero mean uncorrelated Gaussian observation errors with covariance $\mathbf{R}_k$.

With these definitions the EKF-SLAM algorithm is performed with Equations 4 and 5, which gives the following:

**Prediction**

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k), \tag{10}$$

$$\mathbf{P}_{xx,k|k-1} = \mathbf{F}\,\mathbf{P}_{xx,k-1|k-1}\,\mathbf{F}^\top + \mathbf{Q}_k, \tag{11}$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted robot pose, $\mathbf{P}_{xx,k|k-1}$ is the predicted robot pose covariance, and $\mathbf{F}$ is the Jacobian of $f$ evaluated at $\hat{\mathbf{x}}_{k-1|k-1}$, i.e., the estimate up to step $k-1$.

**Measurement Update**

$$\hat{\mathbf{y}}_k = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \hat{\mathbf{m}}_{k-1} \end{bmatrix} + \mathbf{K}_k \left(\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}, \hat{\mathbf{m}}_{k-1})\right), \tag{12}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\,\mathbf{P}_{k|k-1}, \tag{13}$$

where

$$\mathbf{S}_k = \mathbf{H}\,\mathbf{P}_{k|k-1}\,\mathbf{H}^\top + \mathbf{R}_k, \tag{14}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\, \mathbf{H}^\top\, \mathbf{S}_k^{-1}, \tag{15}$$

and where $\mathbf{H}$ is the Jacobian of $h$ evaluated at $\hat{\mathbf{x}}_{k|k-1}$ and the map estimate $\hat{\mathbf{m}}_{k-1}$ up to step $k-1$.

This solution is very well known and inherits many of the same benefits and problems as the standard solutions to navigation or tracking problems. Some of the key issues are the non-linearity of the observation and motion models, filter convergence, data association, and computational effort.

The linearization of the perceptual models introduces optimistic estimations in every iteration, which in the limit produce filter inconsistency. This is specially seen in the case of rotations [6, 120]. Convergence and consistency can only be guaranteed in the linear case. An alternative to the EKF is the use of the UKF (Unscented Kalman Filter) [69], which avoids linearizations through mean and covariance parameterization by particles chosen in a deterministic and geometrical way around the probability distributions. This filter has been also used to solve the SLAM problem [4, 86].

In EKF-SLAM, convergence of the map is manifested in the monotonic convergence of the determinant of the map covariance matrix toward zero. That is, we get full correlation in the map, or said differently, absolute certainty about relations between landmarks. On the other hand, the absolute landmark variances converge to a constant bounded by below by the initial robot position uncertainty. So there is bounded uncertainty of the absolute positions [23]. This is due to the fact that SLAM is a coupled estimation problem, and in consequence is partially observable [3].

The "loop–closure "problem, when a robot returns to re-observe landmarks after a large traverse, is especially difficult. The data association problem is complicated in environments where landmarks are not just points and indeed look different from different view-points, as well as environment with aliasing, that is, regions that are very similar and difficult to distinguish one from the other.

An important advance in the data association problem was the concept of batch gating, where multiple associations are considered simultaneously. Mutual association compatibility exploits the geometric relationship between landmarks. The two existing forms of batch gating are the joint compatibility branch and bound (JCBB) [95] method, which is a tree-search method; and combined constraint data association (CCDA) [6], which is a graph search technique. The former (and also a randomized variant of JCBB [68]) is said to perform reliable association with no knowledge of vehicle pose whatsoever.

Furthermore, the standard formulation of the EKF-SLAM solution is especially fragile to incorrect association of observations to landmarks [18]. In order to validate the history of data association, temporal landmark quality measures and a temporal landmark quality test were proposed in [2]. These quality measures permit the maintenance of the map by the elimination of inconsistent observations. This removal of weak landmarks from the state vector and state covariance matrix did not violate the convergence properties of SLAM.

Other ways to perform reliable data association include other sensing modalities, such as vision. Images provide rich information about shape, color, and texture, all of which may be used to find a correspondence between two data sets. For SLAM, appearance signatures are useful to predict a possible association, such as closing a loop, or for assisting conventional gating by providing additional discrimination information. Historically, appearance signatures and image similarity metrics have been developed for indexing image databases [107] and for recognizing places in topological mapping [5, 126]. Moreover, appearance measures have been applied to detecting loops in SLAM [24, 56, 97]. The work on visual appearance signatures for loop detection by Newman *et al.* [24, 97] introduces two significant innovations. A similarity metric over a sequence of images, rather than a single image, is computed, and an eigenvalue technique is employed to remove common-mode similarity. This approach considerably reduces the occurrence of false positives by considering only matches that are interesting or uncommon. Multihypothesis data association is essential for robust target tracking in cluttered environments [8]. It resolves association ambiguities by generating a separate track estimate for each association hypothesis, creating over time an ever-branching tree of tracks. The number of tracks is typically limited by the available computational resources, and low-likelihood tracks are pruned from the hypothesis tree.

Multihypothesis tracking (MHT) is also important for a robust SLAM implementation, particularly in large complex environments. For example, in loop closure, a robot should ideally maintain separate hypotheses for suspected loops and also a "no-loop" hypothesis for cases where the perceived environment is structurally similar. While MHT has been applied to mapping problems [23], this has yet to be applied in the SLAM context.

Regarding the computational complexity of Kalman Filter methods, we can note that the largest computational complexity of this approach is the measurement update step, which requires that all landmarks and the joint covariance matrix be updated every time an observation is made. The most costly operations in updating the Kalman filter are matrix multiplications which can be implemented in $\mathcal{O}(N^2)$ time, where $N$ is the number of features in the map. When mapping large environments, the computational complexity is a very important issue to solve. There has been a great deal of work undertaken in developing efficient variants of the EKF-SLAM solution.

Techniques aimed at improving computational efficiency may be characterized as being optimal or conservative. Optimal algorithms aim to reduce required computation while still resulting in estimates and covariances that are equal to the full-form SLAM algorithm. Conservative algorithms result in estimates that have larger uncertainty or covariance than the optimal result. Usually, conservative algorithms, while less accurate, are computationally more efficient and, therefore, of value in real implementations. Algorithms with uncertainties or covariances less than those of the optimal solution are termed inconsistent and are considered invalid solutions to the SLAM (or any estimation) problem.

One way to solve the scalability or computational complexity issue is to divide the Kalman filter update step. A number of partitioned update methods have been devised to reduce this computational effort. These confine sensor-rate updates to a small local region and update the global map only at a much lower frequency. These partition methods all produce optimal estimates. There are two basic types of partitioned updates. The first operates in a local region of the global map and maintains globally referenced coordinates. This approach is taken by the compressed EKF (CEKF) [54] and the postponement algorithm [74]. The second generates a short-term submap with its own local coordinate frame. This is the approach of the constrained local submap filter (CLSF) [133] and the local map sequencing algorithm [120].

Submap methods are other means of addressing the issue of computation scaling quadratically with the number of landmarks during measurement updates. Submap methods come in two fundamental varieties: globally referenced and locally referenced. The common thread to both types is that the submap defines a local coordinate frame and nearby landmarks are estimated with respect to the local frame. The local submap estimates are obtained using the EKF-SLAM algorithm using only the locally referenced landmarks. The resulting submap structures are then arranged in a hierarchy leading to computational efficiency but also lack of optimality. Global submap methods estimate the global locations of submap coordinate frames relative to a common base frame. This is the approach adopted in the relative landmark representation (RLR) [55] and hierarchical SLAM [40] methods. These approaches reduce computation from a quadratic dependence on the number of landmarks to a linear or constant time dependence by maintaining a conservative estimate of the global map. However, as submap frames are located relative to a common base coordinate frame, global submaps do not alleviate linearization issues arising from large pose uncertainties.

Relative submap methods differ from global submaps in that there is no common coordinate frame. The location of any given submap is recorded only by its neighboring submaps, and these are connected in a graphical network. Global estimates can be obtained by vector summation along a path in the network. By avoiding any form of global-level data fusion, relative submaps address both computation and nonlinearity issues. The original notion of relative submaps was introduced by Chong and Kleeman [20]. This was further developed by Williams [133] in the form of the constrained relative submap filter (CRSF). However CRSF does not exhibit global-level convergence without forfeiting the decoupled submap structure.

### 3.2.2 Particle Filters

Particle filters can be traced back to [88]. Particle filters represent a posterior through a set of particles. The particle filter has been shown under mild conditions to approach the true posterior as the particle set size goes to infinity. It is a nonparametric representation that represents multimodal distributions with ease. The key problem with the particle filter in the context of SLAM is that the space of maps and robot paths is huge. The approach to make particle filters practical to the SLAM problem goes back to [10, 105]. This approach was introduced into the SLAM literature in [94], followed by [89], who coined the name FastSLAM.

The FastSLAM algorithm solves the full SLAM problem, i.e., it computes estimates for both the set of landmarks positions and the entire robot path. In this method the joint SLAM state is factored into a vehicle component and a conditional map component

$$p(X^k, \mathbf{m}|Z^k, U^k, \mathbf{x}_0) = p(\mathbf{m}|X^k, Z^k)\; p(X^k|Z^k, U^k, \mathbf{x}_0). \tag{16}$$

Here the probability distribution is on the trajectory $X^k$ rather than the single pose $\mathbf{x}_k$ because, when conditioned on the trajectory, the map landmarks become independent. This is a key property of FastSLAM, and the reason for its speed; the map is represented as a set of independent Gaussians, with linear complexity, rather than a joint map covariance with quadratic complexity.

The essential structure of FastSLAM is a Rao-Blackwellised state, where the trajectory is represented by samples $X^k_{(i)}$, wich are weighted by $w^k_{(i)}$, and the map is computed analytically. Thus, the joint distribution, at time $k$, is represented by the set

$$\{w^k_{(i)}, X^k_{(i)}, p(\mathbf{m}|X^k, Z^k)\}^N_i,$$

where the map accompanying each particle is composed of independent Gaussian distributions

$$p(\mathbf{m}|X^k_{(i)}, Z^k) = \prod_j^M p(\mathbf{m}_j|X^k_{(i)}, Z^k).$$

Recursive estimation is performed by particle filtering for the pose states, and the EKF for the map states.

The general form of this method is as follows:

We assume that, at time $k-1$, the joint state is represented by $\{w^{k-1}_{(i)}, X^{k-1}_{(i)}, p(\mathbf{m}|X^{k-1}, Z^{k-1})\}^N_i$.

I. For each particle, compute a proposal distribution, conditioned on the specific particle history, and draw a sample from it

$$\mathbf{x}^{(i)}_k \sim \pi(\mathbf{x}_k|X^{k-1}_{(i)}, Z^k, \mathbf{u}_k). \tag{17}$$

This new sample is (implicitly) joined to the particle history $X_{(i)}^{k} \triangleq \{X_{(i)}^{k-1}\, \mathbf{x}_k^{(i)}\}$.

II. Weight samples according to the importance function.

$$w_{(i)}^{k} = w_{(i)}^{k-1} \frac{p(\mathbf{z}_k | X_{(i)}^{k}, Z^{k-1})\, p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k)}{\pi(\mathbf{x}_k^{(i)} | X_{(i)}^{k-1}, Z^{k}, \mathbf{u}_k)}. \tag{18}$$

The numerator terms of this equation are the observation model and the motion model, respectively. In the observation model the map is marginalized out,

$$p(\mathbf{z}_k | X^{k}, Z^{k-1}) = \int p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m})\, p(\mathbf{m} | X^{k-1}, Z^{k-1})\, d\mathbf{m}. \tag{19}$$

III. If necessary, resampling is performed . Resampling is accomplished by selecting particles, with replacement, from the set $\{X_{(i)}^{k}\}_i^N$, including their associated maps, with probability of selection proportional to $w_{(i)}^{k}$ . Selected particles are given uniform weight, $w_{(i)}^{k} = \frac{1}{N}$.

IV. For each particle, perform an EKF update on the observed landmarks as a simple mapping operation with known vehicle pose.

The two versions of FastSLAM in the literature, FastSLAM 1.0 [89] and FastSLAM 2.0 [90], differ only in terms of the form of their proposal distribution (step I) and, consequently in their importance weight (step II). FastSLAM 2.0 is by far the more efficient solution.

Statistically, FastSLAM (1.0 and 2.0) suffer degeneration due to its inability to forget the past. Marginalizing the map in Equation 19 introduces dependence on the pose and measurement history, and so, when resampling depletes this history, statistical accuracy is lost [7]. Nevertheless, empirical results of FastSLAM 2.0 in real outdoor environments [90] show that the algorithm is capable of generating an accurate map in practice.

FastSLAM has been extended in great many ways. One important variant is a grid-based version of FastSLAM, in which the Gaussians are replaced by an occupancy grid map [57]. Other significant extensions of the FastSLAM method can be found in [38, 39], whose methods DP-SLAM and ancestry trees provide efficient tree update methods for grid-based maps.

### 3.2.3 Graph-Based SLAM

A third family of algorithms consider the so-called “Graph-Based”, “Network-Based”, or “Belief Net” approach in which the SLAM problem is represent as a Dynamic Bayes Network, where landmarks and robot locations can be thought of as nodes in the graph. Every consecutive pair of locations is tied together by an arc that represents the information conveyed by the odometry reading. Other arcs exist between locations and landmarks to represent the observed landmarks at the corresponding location. Arcs in this graph are soft constraints. Relaxing these constraints yields the robot’s best estimate for the map and the full path.

The construction of the graph is illustrated in Figure 1. Suppose that at time $k = 1$ the robot senses landmark $m_1$ . This adds an arc in the (yet highly incomplete) graph between $x_1$ and $m_1$ . When representing the edges in a matrix format (which happens to correspond to a quadratic equation defining the resulting constraints), a value is added to the elements between $x_1$ and $m_1$, as shown in Figure 1a.

Now suppose the robot moves. The odometry reading $u_2$ leads to an increment in the matrix representation and an arc between nodes $x_1$ and $x_2$, as shown in Figure 1b. Consecutive

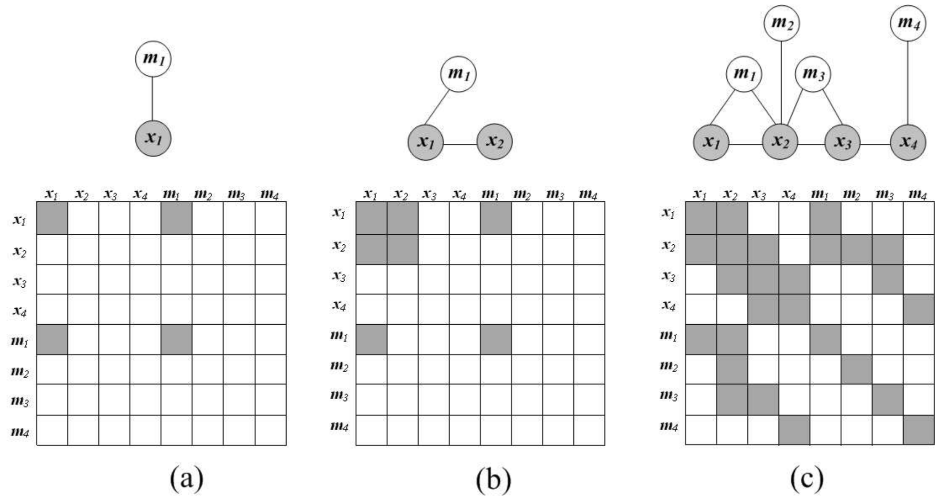


**Figure 1:** Illustration of the graph construction. The upper part of the figure shows the graph and the lower part corresponds to the matrix representation of the constraints

application of these two basic steps leads to a graph of increasing size, additionally the matrix representation increases its size to include the new sensed landmarks and the new robot poses, as illustrated in Figure 1c. Nevertheless this graph is sparse, in that each node is only connected to a small number of other nodes.

Many of these algorithms can be seen as a spring-mass model, where computing the SLAM solution is equivalent to computing the state of minimal energy of this model. To see this, we can note that the graph corresponds to the log-posterior of the full SLAM problem [123]:

$$\log p(X^k, \mathbf{m}|Z^k, U^k). \tag{20}$$

This logarithm is of the form

$$\log p(X^k, \mathbf{m}|Z^k, U^k) = c + \sum_k \log p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) + \sum_k \log p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m}), \tag{21}$$

where $c$ is a constant.

Each constraint of the form log $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)$ is the result of exactly one robot motion event, and it corresponds to an arc in the graph. Likewise, each constraint of the form $\log p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{m})$ is the result of one sensor measurement, to which we can also find a corresponding arc in the graph. In this case, the SLAM problem is to find the mode of this equation,

$$\hat{X}^k, \hat{\mathbf{m}} = \underset{X^k, \mathbf{m}}{\arg\max} \log p(X^k, \mathbf{m}|Z^k, U^k). \tag{22}$$

Under the Gaussian assumption, this expression resolves to the following quadratic form

$$\begin{aligned} \log p(X^k, \mathbf{m}|Z^k, U^k) = \quad & c + \textstyle\sum_k [\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)]\ \mathbf{Q}_k^{-1}\ [\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)]^\top \\ & + \textstyle\sum_k [\mathbf{z}_k - h(\mathbf{x}_k, \mathbf{m})]\ \mathbf{R}_k^{-1}\ [\mathbf{z}_k - h(\mathbf{x}_k, \mathbf{m})]^\top. \end{aligned} \tag{23}$$

This is a sparse function, and to solve it, a number of efficient optimization techniques can be applied. Common choices include gradient descent, conjugate gradient, and others. Most SLAM implementations rely on some sort of iterative linearizing the functions $f$ and $h$, in which case the objective in Equation 23 becomes quadratic in all of its variables.

Many of these techniques address the full SLAM problem and are inherently offline, that is, they optimize for the entire robot path. If the robot path is long, the optimization may become cumbersome. However, there are some that incrementally solve the full SLAM problem. Aditionally, some solutions deal with the online SLAM or the trajectory-oriented SLAM.

Graph-Based SLAM methods have the advantage that they scale to much higher-dimensional maps than EKF SLAM. As we already mentioned, the key limiting factor in EKF-SLAM is the covariance matrix, which takes space (and update time) quadratic in the size of the map; however, in this paradigm no such constraint exists. Additionally, the Network-Based paradigm is easily extended to handle the data association problems [59, 81].

Graph-Based techniques were first mentioned in [36, 116], but the seminal paper of Lu and Milios [83] provided a first working solution. Their approach computes maximum likelihood maps by least square error minimization. The idea is to compute a network of relations given the sequence of sensor readings. These relations represent the spatial constraints between the poses of the robot. Their approach seeks to optimize the whole network at once. They used scan matching, that is, scans are localized relative to slightly older scans and, once localized, are added to the map under the assumption that the estimated location is correct.

Gutmann and Konolige [56] proposed an effective way for constructing a network of constraints and for detecting loop closures while running an incremental estimation algorithm. Howard *et al.* [64] apply relaxation to localize the robot and build a map. Duckett *et al.* [34] described an implementation that uses Gauss-Seidel relaxation. In order to make the problem linear, they assume knowledge about the orientation of the robot. Frese *et al.* [49] proposed a variant of Gauss-Seidel relaxation called multi-level relaxation (MLR). It applies relaxation at different resolutions. Multi-resolution methods are typically applied to problems more spatially uniform than that of SLAM, but they report good results.

Other nonlinear approaches include GraphSLAM [123], and Graphical SLAM [45]. Additionally, Konolige proposed a method [76] for accelerating convergence by reducing the graph to poses that have a loop constraint attached, solving for the other nodes separately. This can save considerable CPU time, but requires the graph to have low connectivity.

Paskin's Thin Junction Tree Filter [104] and Frese's TreeMap [48] compute nonlinear map estimates. They require factorization of the joint probability density, which they achieve by ignoring small state correlations. Frese's TreeMap works by hierarchically dividing a map into local regions and subregions. It uses a binary-tree to perform integration and marginalization, leading to an impressive algorithm that has the ability to deal with very large maps. On the other hand, once variables are combined in a node, they cannot be separated again. In this way, several different linearization points are propagated in different factors leading to inconsistency of the final result. Aditionally, since multiple approximations are employed to reduce the complexity, it results in noticeable map artifacts. Although this approach let us perform very fast inference, we can not recover the cross correlations to recover the covariance matrices needed to do data association.

A hybrid of linear and nonlinear solutions is Bosse's Atlas [12], which uses linearized (EKF-based) submaps but merges them together using nonlinear optimization. Nonlinear optimization algorithms have a rich history outside the SLAM community. SLAM algorithms have typically limited themselves to Gauss-Seidel or Gradient Descent approaches, but other approaches are commonly used in other fields. In particular, Stochastic Gradient Descent is often used to train artificial neural networks. To solve the SLAM problem, Olson *et al.* [103] presented a fast non-linear optimization algorithm based on a variant of Stochastic Gradient Descent on an alternative state-space representation. The proposed Stochastic Gradient Descent variant is robust against local minima and converges quickly. Moreover, the alternative state space representation allows a single iteration to update many poses without incurring a large computational cost. Some extensions of this approach were presented by Grissetti *et al.*[51, 52, 53]. Grissetti's approach uses a tree structure to define and efficiently update local regions in each iteration. The poses of the individual nodes are represented in an incremental fashion which allows the algorithm to automatically update successor nodes, which converges significantly faster than Olson's algorithm. Although these methods let us avoid linearization errors and recover the robot trajectory efficiently, they do not consider the data association problem, therefore, they need to be used together with other methods to obtain the necessary data associations.

The Graph-Based paradigm is very closely linked to Information Filter methods. In contrast to Extended Kalman Filters, Information Filters employ the information form of the Gaussian distribution. The information form is often called the canonical or natural representation of the Gaussian distribution. This notion of a natural representation stems from expanding the quadratic in the exponential of the Gaussian distribution as

$$\begin{aligned}
p(\mathbf{x}_k) &= \mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
&= \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_k)^\top\right\} \\
&= \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}_k^\top\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_k - 2\boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_k + \boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k\right)\right\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k}}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}} \exp\left\{-\frac{1}{2}\mathbf{x}_k^\top\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_k + \boldsymbol{\mu}_k^\top\boldsymbol{\Sigma}_k^{-1}\mathbf{x}_k\right\} \\
&= \frac{e^{-\frac{1}{2}\boldsymbol{\eta}_k^\top\boldsymbol{\Lambda}_k^{-1}\boldsymbol{\eta}_k}}{\sqrt{|2\pi\boldsymbol{\Lambda}_k^{-1}|}} \exp\left\{-\frac{1}{2}\mathbf{x}_k^\top\boldsymbol{\Lambda}_k\mathbf{x}_k + \boldsymbol{\eta}_k^\top\mathbf{x}_k\right\} \\
&= \mathcal{N}^{-1}(\mathbf{x}_k; \boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)
\end{aligned} \tag{24}$$

where

$$\boldsymbol{\Lambda}_k = \boldsymbol{\Sigma}_k^{-1},$$

and

$$\boldsymbol{\eta}_k = \boldsymbol{\Lambda}_k\boldsymbol{\mu}_k.$$

Therefore, the Information Filter rather than parameterizing the normal distribution in terms of its mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$, $\mathcal{N}(\mathbf{x}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, they use instead a parametrized Gaussian in terms of its information vector $\boldsymbol{\eta}_k$ and information matrix $\boldsymbol{\Lambda}_k$, $\mathcal{N}^{-1}(\mathbf{x}_k; \boldsymbol{\eta}_k, \boldsymbol{\Lambda}_k)$.

The covariance and information representations lead to very different computational characteristics with respect to the fundamental probabilistic operations of marginalization and conditioning. This is important, because these two operations appear at the core of any SLAM algorithm (e.g. in Prediction and Measurement Update). The covariance and information representations exhibit a dual relationship with respect to marginalization and conditioning. Marginalization is easy in the covariance form, since it corresponds to extracting the appropriate subblock from the covariance matrix, while in the information form, it is hard, because it involves calculating the Schur complement over the variables we wish to keep. The opposite relation holds true for conditioning, which is easy in the information form and hard in the covariance form.

One quality of the canonical form is its relationship with Gaussian Markov Random Fields in which nodes in the graph represent individual state variables and edge structure describes their conditional independence relationships. The information matrix serves as an adjacency matrix for the graph representation, with the strength of constraints between pairs of variables proportional to the corresponding elements of the matrix and zero components denote the absence of links in the Bayes Network.

Thrun *et al.* [125] made the empirical observation that the normalized information matrix obtained when the SLAM problem is formulated in the information form is approximately sparse, i.e. entries of distant landmarks are very small and thus can be replaced by a sparse approximation. In [125] they utilize this observation in their Sparse Extended Information Filter (SEIF). Paskin's Thin Junction Tree Filter [104] and Frese's TreeMap [48] were also based on this observation. Theoretical explanation for this observation was later presented by Frese [47]. Sparsification essentially removes the weak links in the information matrix by setting elements that are smaller than a given threshold to zero, while strengthening other links to make up for the effect of this change. With the information matrix now sparse, very efficient update procedures for information estimates can be obtained.

On the other hand, Eustice *et al.* [43] showed that the SEIF method causes inconsistent estimates. They showed that map inconsistency occurs within the global reference frame where estimation is performed. In [43] they also proposed a slightly modified version of the Thrun's sparsification procedure, which is shown to preserve sparsity while also generating both local and global map estimates comparable to those obtained by the non-sparsified SLAM filter. However this modified approximation is no longer constant-time. Eventually, Eustice *et al.* [41] showed that if only the robot trajectory is estimated, the associated information is exactly sparse. This fact was used in the Exactly Sparse Delayed-State Filter (ESDF) that they proposed. In open loop navigation, the information matrix becomes tri-block diagonal as consecutive robot poses are added to the state. At loop closure, the matrix structure is only modified sparsely, setting information links between non-consecutive robot poses. In the ESDF method predictions and updates take constant time, assuming an efficient or approximate way for state recovery is used to evaluate Jacobians. It has been noted in [131] that the ESDF method is suitable for the scenarios where features are difficult to extract or the number of features is too large as compared with robot poses.

A solution for the online SLAM problem which yields an exactly sparse information matrix was proposed by Walter *et al.* [128, 129]. Their method was called Exactly Sparse Extended Information Filter (ESEIF) and achieves sparsity by periodically marginalizing out and relocating robot. Similar to SEIF, ESEIF exploits the fact that when the robot location is marginalized out from the state vector, new links will only be built up among the features that were previously linked with the robot in the information matrix. The set of features that are linked with the robot is called “active features”. Thus the information matrix will be sparse if the number of “active features” is bounded. In contrast to the sparsification process in SEIF, ESEIF controls the number of “active features” by “kidnapping” the robot when the number of “active features” is about to become larger than a predefined threshold. This is followed by “relocating” the robot using a set of selected measurements. Thus, the information matrix is kept sparse without any approximations that can lead to inconsistency. The extent of sparseness is controlled by the active feature bound, the sensor range and the feature density. However, there is some information loss in ESEIF due to “kidnapping” and “relocating” the robot.

A method similar to ESEIF is D-SLAM, proposed by Wang *et al.* [130], which has a similar state vector and does not use any approximations to achieve sparseness. D-SLAM uses a state vector that only contains the feature locations to generate maps of an environment. The robot location estimate is obtained through a concurrent yet separate process. In D-SLAM mapping, the original measurements relating the robot and features are first transformed into relative distances and angles among features. Then these transformed measurements are fused into the map using an Extended Information Filter (EIF). It is shown that only the features that are observed at the same time instant have links in the information matrix making it exactly sparse. The extent of sparseness is governed by the sensor range and feature density. Localization is performed by combining two estimates: one is obtained by solving a "kidnapped robot problem"; the other is obtained by a local EKF SLAM where only the features currently observed are retained in the state vector. The two correlated estimates are fused by Covariance Intersection (CI). Exact state and covariance recovery is achieved by preconditioned Conjugated Gradient (PCG). A good preconditioner produced by an iterative Cholesky factorization method exploiting the similarity between the information matrices of successive steps is used to make the PCG efficient. Data association is solved by a combination of the standard maximum likelihood approach and a Chi-Square test. However, there is also some information loss in D-SLAM.

A drawback of the information filter techniques is that we no longer have access to the mean vector when the posterior is represented in the canonical form. Inverting the information matrix is cubic in the number of states, making it intractable, even for small maps. However, recovering the mean can be achieved by solving a set of linear equation taking advantage of the sparseness of the information matrix. There are a number of techniques that iteratively solve such sparse, symmetric positive definite systems of equations including conjugate gradient descent as well as relaxation-based algorithms such as Gauss-Seidel , or multilevel method [49, 76]. The optimizations can often be performed over the course of multiple time steps since, aside from loop closures, the mean vector evolves slowly in SLAM. As a result, we can fix the number of iterations per update [34, 125].

On the other hand, if we are only interested in a subset of the mean such as during the time projection step we can then consider partial mean recovery as proposed by Eustice *et al.* [41]. Recently, Huang *et al.* [65] proposed a submap based Exactly Sparse Extended Information Filter SLAM algorithm, called "Sparse Local Submap Joining Filter" (SLSJF), which achieves exact recovery instead of approximate recovery. The recovery algorithm is based on an incremental Cholesky factorization approach and a natural reordering of the global state vector.

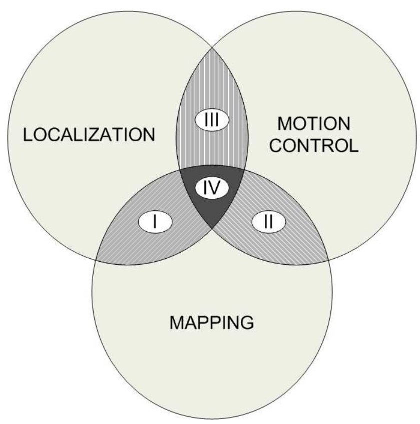
Another approach that makes use of sparse information matrices is the "square root SAM" proposed by Dellaert [30, 31], which is based on Square-Root Filtering [8]. This approach seeks the "Maximum a Posteriori" estimate for the robot poses and the map (Equation 22). Given that this method solves the full SLAM problem, the information matrix becomes naturally sparse, since no robot pose marginalization is performed. The "square root SAM" factorizes the information matrix or the measurement Jacobian into square root form, using sparse Cholesky or QR factorization, respectively, yielding a square root information matrix that can be used to obtain the optimal robot trajectory and map. On the other hand, since is smoothed the entire trajectory, computational complexity grows without bound over time. In addition, this method does not consider the data association problem and it has to relinearize the measurement equations and re-factorize at every iteration. Nonetheless, this approach was developed further by Dellaert's research group.

Ni *et al.* [99] proposed a method called Tectonic SAM, which combines a submap-based approach with"square root SAM". It solves the computational burden of re-factoring and relinearizing at every iteration by introducing base nodes to capture the global positions of submaps. The algorithm first solves each submap locally by updating the linearization using the new estimation in each iteration. Then a global alignment follows to compute the relative location of the submaps. Given the fact that the submaps will only slightly change after incorporating all local constraints, the linearization points of the submap variables can be fixed in this stage. As a result, most linearization calculations need to be done only in local submaps while they can be kept constant after that.

Kaess *et al.* [71] introduced an algorithm called "incremental Smoothing and Mapping" (iSAM), which is an incremental smoothing approach that takes into account the data association problem. This combines the advantages of factorization-based "square-root SAM" with real-time performance for adding new measurements and obtaining the trajectory and the map. This method allows access to the exact marginal covariances, without having to calculate the full covariance matrix. This approach relies on a QR factorization of the information matrix and integrates the new measurements as they are available. Using the QR factorization, the poses of the nodes in the network can be retrieved by back substitution. Moreover, they keep sparse matrices by occassinal variable reordering.

### 3.2.4 Active SLAM

The solutions to the SLAM problem considered so far are passive, in the sense that they only process incoming sensor data without explicitly generating controls. In this section, we consider the problem of active SLAM, also referred as integrated exploration, or robot exploration for SLAM. In this problem it is assumed that both the robot's pose and the map are unknown; therefore, the robot must need to find the appropriate motion commands in order to reduce the uncertainty about its own pose and the map. These techniques jointly optimize for map coverage and active localization. In Table 1 we show the different branchs of robotic exploration, as was presented in [84]. The problem of active SLAM is shown in the region IV of Table 1.




I. Passive solutions for SLAM
II. Classic Exploration
III. Active Localization
IV. active SLAM

**Table 1:** The field of Robotic exploration. The overlapping areas represent combinations of the mapping, localisation, and control problems.

The key insight for optimal exploration in SLAM is that the entropy of the SLAM posterior can be decomposed into the entropy in the pose posterior and into the expected entropy of the map [123]. In this way, a robot performing active SLAM trades off uncertainty in the robot pose with uncertainty in the map. When closing a loop, a robot will mainly reduce its pose uncertainty, however, when the robot moves into an unexplored region, it will mainly reduce its map uncertainty. By considering both, whatever reduction is larger will win, and the robot may sometimes move into unknown terrain or sometimes re-localize by moving into known terrain.

Considerable work has been done in the field of Partially Observable Markov Decision Processes (POMDPs). However, one drawback to the use of traditional POMDPs to exploration problems is that they can become computationally intractable. The Coastal Navigation method proposed by Roy *et al.* [106] introduced the POMDPs formalism to the exploration problem through simplifying assumptions in order to reduce the complexity. Moreover, this approach reduces the entropy and includes the task goal into the planner's objective function.

In the context of exploration for SLAM, early contributions can be traced back to the work of Feder, Leonard and Smith [44]. In their work, the vehicle creates a map and localizes simultaneously and makes local decisions on where to move next in order to minimize the error in estimates of the vehicle pose and the landmark locations. The authors discussed how the robot dealt with the task of localization and mapping in an adaptive way, and gave the specific expression to quantitatively explain the uncertainty of the robot and environment features. The technique mentioned has been demonstrated via simulations, sonar experiments, and underwater sonar experiments. This principle is applied to the problem of underwater exploration in the work of Bennett *et al.* [9]. The motion command to minimize the vehicle pose and map error is incorporated into a general behavior based architecture. In both cases the exploration approach is spatially local and greedy, that is, they chose control by maximizing information is based on a single-step look-ahead.

In [14] an also greedy adaptive SLAM approach based on EKF is proposed, which implemented active planning by selecting the best control input that will get the highest accuracy on the next step. This approach made the assumption that the robot would observe all the landmarks any time for simplicity, which limited the approach to be applied in large environments. The same authors introduced in [84] a utility function which trades-off the cost of reaching frontiers with the utility of selected positions with respect to a potential reduction of pose uncertainty.

In [118], Stachniss *et al.* conducted frontier based exploration [134] with SLAM. Using two maps, occupancy grid and topological, active loop-closing was performed. Their algorithm used a grid-based version of the FastSLAM method and explicitly takes into account the uncertainty about the pose of the robot during the exploration tasks. Additionally, it avoids overconfident in robot pose when actively closing loops, a typical problem of particle filters in this context. However, maximization of information gain along the path was not considered. Newman *et al.* [98] proposed an exploration approach in the context of Bosse's ATLAS [11]. Here the robot builds a graph-structure to represent visited areas, and planned motion is motivated by the geometric, spatial, and stochastic characteristics of the current map. Each feature within the map is responsible for determining nearby unexplored areas. They assumed that the location of the features is uncertain and represented by a set of probability functions, which are used in conjunction with the robot path history to determine a robot trajectory suited for exploration.

Sim *et al.* [112] proposed to encourage coverage by randomly placing dummy features in unexplored areas. A Voronoi graph was used for path planning, with assumptions of perfect data association and unlimited sensor field of view. Yet, this strategy is not effective for systems with short planning horizons and limited sensing as the dummy features will not influence the robot's decision if they are not visible within the planning horizon. In [114] Sim conducted active SLAM with a grid based approach and showed that optimisation using the trace of the covariance matrix from the EKF performs better than using the determinant. However it is assumed that the robot is quasi-holonomic with unlimited sensing and robot motion constraints are not considered in the planning process. In [111], Sim addresses the stability issue. Features that are too close to the robot which may cause filter instability are blocked by using a virtual minimum range sensor.

Davison *et al.* [27] proposed a SLAM solution using active vision. They controled a stereo-head considering uncertainty-based measurement selection, automatic map-maintenance, and goal-directed steering. However, they only control orientations of the stereo-head. Vidal *et al.* [127] considered a single hand-held camera performing SLAM at video rate with generic 6 DOF. They optimized both the localization of the sensor and building of the feature map by choosing a maximal mutually informative motion command. In [15], Bryson *et al.* presented simulated results of the effect different vehicle actions have with respect to mutual information gain. The analysis is performed for a 6 DOF aerial vehicle equipped with two cameras and an inertial sensor, for which landmark range, azimuth, and elevation readings are simulated, and data association is known.

Recently, Leung *et al.* [79] considered the trajectory planning problem for line-feature based SLAM in structured indoor environments. The robot poses and line features are estimated using iSAM [71], given that it is found to provide more consistent estimates than the Extended Kalman Filter. The trajectory planning seeks to minimize the uncertainty of the estimates and to maximize coverage. Trajectory planning is performed using Model Predictive Control (MPC) with an attractor incorporating long term goals. However, they are restricted to small indoor planar environments.

### 3.2.5 Environment Representation

The firts works in SLAM modeled the world as a set of simple discrete landmarks described by geometric primitives such as points, lines, or circles. In more complex and unstructured environments –outdoor, underground, subsea– this assumption often does not hold. Implementing SLAM in three dimensions is, in principle, a straightforward extension of the two-dimensional case. However, it involves significant added complexity due to the more general vehicle motion model and feature modeling complexity.

While EKF-SLAM was usually applied to geometric landmarks (often point landmarks), attaching a coordinate frame to an arbitrary object allows the same methods to be applied to much more general landmark descriptions. Castellanos *et al.* [16] proposed a framewok, called "Symmetries and Perturbations Model" (SPmodel), for representing and processing erroneous geometrical data. Within this framework, the location of a geometrical object is defined by its position and orientation in 3D from the world coordinate frame into a local object coordinate frame, where a locally defined perturbation vector represents the error with its associated covariance matrix containing the uncertainty information.

Hähnel *et al.* [58], Thrun *et al.* [122], Liu *et al.* [80], Moravec *et al.* [92], and Surmann *et al.* [119] have also proposed extension of SLAM to three dimensions. Horn *et al.* [63] corrected the planar (2D) pose of a mobile robot by using vertical planes extracted from 3D data. Sequeira *et al.* [110] used a single point laser mounted on a pan-tilt unit to create 3D models of indoor scenes. Their work mainly focussed on next-view planning and 3D reconstruction. Nüchter *et al.* [102] presented a scan alignment approach for 3D scans gathered by a rotating laser scanner sensor. They use the iterative closest point (ICP) algorithm to minimize a global error measure to generate consistent models. The approach presented by Weingarten *et al.* [132] uses planar features extracted probabilistically from dense three-dimensional point clouds generated by a rotating 2D laser scanner. These features are represented in compliance with the Symmetries and Perturbation model (SPmodel) in a stochastic map. Kohlhepp *et al.* [75] also extracted planes to track the robot pose along with an EKF-based approach but represented the map in separate submaps. In this approach a single consistent stochastic map is used to represent the robot pose and all features. Cole and Newman [22] proposed a 3D scan approach within a delayed-state framework. They used a segmentation algorithm to separate the data stream into distinct point clouds, each referenced to a vehicle position. The SLAM technique they adopted inherits much from 2D delayed-state SLAM in that the state vector is an ever growing stack of past vehicle positions and inter-scan registrations are used to form measurements between them.

Apart from scan data for mapping, it is possible to associate auxiliary information with each pose, soil salinity, humidity, temperature, or terrain characteristics, for example. The associated information may be used to assist mapping, to ease data association, or to aid path planning (e.g. traversability maps). This concept of embedding auxiliary data is more difficult to incorporate within the traditional SLAM framework. Therefore, trajectory-oriented SLAM lends itself to representing this information spatially located. Nieto *et al.* [100] have devised a method called DenseSLAM to permit such an embedding. As the robot moves through the environment, auxiliary data is stored in a suitable data structure, such as an occupancy grid, and the region represented by each grid cell is determined by a set of local landmarks in the SLAM map. As the map evolves, and the landmarks move, the locality of the grid region is shifted and warped accordingly. The result is an ability to consistently maintain spatial locality of dense auxiliary information using the SLAM landmark estimates.

Real-world environments contain moving objects, such as people, and temporary structures that appear static for a while but are later moved, such as chairs and parked cars. In dynamic environments, a SLAM algorithm must somehow manage moving objects. It can detect and ignore them; it can track them as moving landmarks, but it must not add a moving object to the map and assume it is stationary. The conventional SLAM solution is highly redundant. As noted in [2] landmarks can be removed from the map without loss of consistency, and it is often possible to remove large numbers of landmarks with little change in convergence rate. This property has been exploited to maintain a contemporaneous map by removing landmarks that have become obsolete due to changes in the environment. To explicitly manage moving objects, Hähnel *et al.* [25] implement an auxiliary identification routine and then remove the dynamic information from a data scan before sending it to their SLAM algorithm. Conversely, Wang *et al.* [51] add moving objects to their estimated state and provide models for tracking both stationary and dynamic targets. Simultaneous estimation of moving and stationary landmarks is very costly due to the added predictive model. For this reason, the implemented solution first involves a stationary SLAM update followed by separate tracking of moving targets.

Other environment representations rely mainly on visual information. This kind of solutions are referred as Visual SLAM. Mapping with cameras has some advantages, they are low cost, light, and power-saving. These sensors can perceive data in a volume, very far, and very precisely. In addition, images carry a vast amount of information and a vast know-how exists in the computer vision community. Recovering the geometry of a 3D scene has been studied for years in computer vision and photogrammetry. Several approaches, such as Structure From Motion (SFM) methods, attempt to extract the scene geometry from a sequence of two-dimensional images. SFM problems are a subset of the more general vision shape recovery problem. SFM seeks to take the 2D information and recover the original 3D information, inverting the effect of the projection process. Bundle adjustment (BA) is frequently used to improve upon SFM solutions [60]. BA is the problem of improving a visual reconstruction to produce both optimal structure and optimal viewing parameter estimates. Assuming a Gaussian noise model, BA can be equated with a maximum likelihood estimator. An error minimization is performed numerically, typically using a non-linear least squares method. Moreover, Structure From Motion in Computer Vision and Simultaneous Map Building and Localization for mobile robots are two views of the same problem: estimation of the motion of a body which moves through a static environment about which it has little or no prior knowledge, using measurements from its sensors to provide information about its motion and the structure of the world. This body could be a single camera, a robot with various sensors, or any other moving object able to sense its surroundings [26].

In the context of SLAM, Davison and Murray [27, 25, 28] proposed a direct extension of 2D EKF-SLAM to three dimensions, with the extraction of discrete visual landmarks and joint estimation of the map and vehicle pose. This work was the first visual SLAM system with processing in real time able to build a 3D map of natural landmarks on the fly and control a mobile robot. Davison and Kita [26] extended this method to the case of a robot able to localize while traversing nonplanar ramps by combining stereo vision with an inclinometer. Davison's work on SLAM with a single camera (MonoSLAM) [29] showed that is possible to build a sparse map of high quality features in real-time.

There has been significant extensions of Davison's original EKF-SLAM based algorithm. Davison's feature initialization scheme using an auxiliary particle filter was improved on by

Solà *et al.* [117] with a mixture of Gaussians method, and then by Eade and Drummond [37] and Montiel *et al.* [91] with a new inverse depth parameterization which can seamlessly cope with features at any depth, and is able to work without any known initial pattern in the scene. Civera *et al.* [21] proposed the use of a hybrid estimation scheme, implementing a fully Bayesian Interacting Multiple Models (IMM) framework which can switch automatically between parameter sets in a dimensionless formulation of monocular SLAM. Holmes *et al.* [62] showed how the Squared Root Unscented Kalman Filter (SRUKF) for the SLAM problem, using a single camera, can be re-posed with $\mathcal{O}(N^2)$ complexity, matching that of the EKF. Jung and Lacroix [70] presented a stereo vision SLAM system using a downward-looking stereo rig to localize a robotic airship and perform terrain mapping. Kim and Sukkarieh [73] used monocular vision in combination with inertial sensing to map ground-based targets from a dynamically maneuvering Unmanned Autonomous Vehicle (UAV).

Bosse *et al.* [11, 13] used omnidirectional vision in combination with other sensors in their ATLAS submapping framework. Eustice *et al.* [42] used a single downward-looking camera and inertial sensing to localize an underwater remote vehicle and produce detailed seabed reconstructions from monocular camera sequences, using an information filter within a Delayed Sate framework. Ila *et al.* [66, 67] used a similar approach, however they used SIFT features [82, 109] matches over consecutive pairs of images for the computation of 6D relative pose constraints (3D position and Euler angles) instead of the use of combined Harris and SIFT matches over monocular camera sequences for the computation of 5 degrees-of-freedom (DOF) relative orientation constraints (azimuth, elevation and Euler angles). The commercial vSLAM system [72] also uses SIFT features, though within a SLAM algorithm which relies significantly on odometry to build a connected map of recognizable locations rather than fully continuous accurate localization.

Sim *et al.* [113] used an algorithm combining SIFT features and FastSLAM filtering to achieve particularly large-scale vision-only SLAM mapping. Marks *et al.* [85] also used a particle filter approach to stereo visual SLAM. They obtain a joint posterior over poses and maps using a Rao-Blackwellized particle filter. The pose distribution is estimated using the particle filter, and each particle has its own map that is obtained through exact filtering conditioned on the particle's pose.

McLauchlan and Murray [87] introduced the VSDF (Variable State-Dimension Filter) for simultaneous structure and motion recovery from a moving camera using a sparse information filter framework, but were not able to demonstrate long-term tracking or loop closing. The approach of Chiuso *et al.* [19] used a single Extended Kalman Filter to propagate the map and localization uncertainty, but only limited results of tracking small groups of objects with small camera motions were presented. Their method used simple gradient descent feature tracking and was therefore unable to match features during high acceleration or close loops after periods of neglect. Foxlin [46] used fiducial markers attached to the ceiling in combination with high-performance inertial sensing. This system achieved very impressive and repeatable localization results, but with the requirement for substantial extra infrastructure and cost. Burschka and Hager [39] demonstrated a small-scale visual localization and mapping system, though by separating the localization and mapping steps they neglect estimate correlations.

Nister *et al.* [101] presented a real-time Visual Odometry system based on the standard structure from motion methodology of frame-to-frame matching of large numbers of point features which was able to recover instantaneous motions impressively but again had no ability to recognize features after periods of neglect and, therefore, would lead inevitably to rapid drift in augmented reality or localization. Konolige *et al.* [77] proposed a Visual Odometry over long courses in natural terrain, where wheel based odometry is less reliable. They proposed the used of a new multiscale feature called CenSurE [1] which was reported to have stability over both outdoor and indoor sequences, and to be inexpensive to compute.

### 3.2.6 Conclusions

We presented a literature survey of solutions to the SLAM problem, considering the estimation process, autonomous exploration, and the environment representation. Regarding the estimation process, we noted that the literature has identified three main paradigms: EKF-SLAM, Particle Filters, and Graph-Based approaches. EKF-SLAM comes with a computational hurdle that poses serious scaling limitations. The most promising extensions of EKF-SLAM are based on building local submaps; however, in many ways the resulting algorithms resemble the graph-based approach.

Particle Filter methods sidestep some of the issues arising from the natural inter-feature correlations in the map, which plagued the EKF. By sampling from robot poses, the individual landmarks in the map become independent, and hence are decorrelated. As a result, FastSLAM can represent the posterior by a sampled robot pose, and many local, independent Gaussians for its landmarks. The particle representation of FastSLAM has a number of advantages. Computationally, FastSLAM can be used as a filter, and its update requires linear-logarithmic time where EKF needed quadratic time. Further, FastSLAM can sample over data association, which makes it a prime method for SLAM with unknown data association. However, a scale problem also afflicts Particle Filters approaches. The number of necessary particles can grow very large, especially for robots seeking to map multiple nested loops and it is unclear how many particles are necessary for a given environment. Another difficulty arising in Particle Filter alternatives to the SLAM problem is that a theoretical proof of asymptotic convergence in estimation is yet to be shown.

The Graph-Based methods draw their intuition from the observation that SLAM can be modeled by a sparse graph of soft constraints, where each constraint either corresponds to a motion or a measurement event. Due to the availability of highly efficient optimization methods for sparse nonlinear optimization problems, Graph-Based SLAM has become the method of choice for building large-scale maps offline. Data association search is quite easily incorporated into the basic mathematical framework, and a number of search techniques exist for finding suitable correspondences. Many Graph-Based methods address the full SLAM problem, and hence are by nature not online. On the other hand, we showed that works such as the iSAM method [71] iteratively solve the full SLAM problem. Moreover, we showed that there are some works in this paradigm that consider other formulations, such as the online SLAM or the trajectory-oriented SLAM. Some of them exploit sparse information matrices and can achieve near constant time performance.

In passive SLAM algorithms, some other entity controls the robot, and the SLAM algorithm is purely observing. In these methods the robot designer has the freedom to implement arbitrary motion controllers, and pursue arbitrary motion objectives. As we noted, in active SLAM the robot actively explores its environment in the pursuit of an accurate localization and mapping. As opposed to classic robot exploration, in this problem the robot pose and the map is unknown, hence active SLAM jointly optimizes for map coverage and active localization (as shown in Table 1). There exist hybrid techniques in which the SLAM algorithm controls only the pointing direction of the robot sensors, but not the motion direction. The majority of the works use an existing passive SLAM approach in conjunction with a method that greedely selects motion commands based on an information gain measure. One popular measure of information gain of probability distributions is relative entropy. Yet globally optimal exploration in SLAM is still and open problem

In the literature review we also classified works by their environment representation. This kind of classification is more related to implementation issues and sensor modalities; therefore, the reviewed works show how the previous discussed approaches are applied. As we noted, early works in SLAM modeled the environment as a set of simply discrete landmarks, however, given that many environments lack of structure, this kind of environment modeling may be insufficient. Some representations include auxiliary information about the environment, other include moving objects as well as arbitrary shaped landmarks. Other works implement SLAM in three dimensions. As we mentioned, it may seem that 3D SLAM is a straightforward extension of the two-dimensional case, however, it involves significant added complexity due to the vehicle motion model, sensor modalities, and feature modeling.

In Table 2 we summarize the most illustrative works on SLAM mentioned in this literature survey. The field labeled *Formulation* refers to the formulation used (online SLAM, full SLAM, or trajectory-oriented SLAM). The item *Data Association* specifies whether an algorithm can cope with unknown correspondence problems. The field *Incremental* tell us whether maps can be built incrementally (and possibly in real-time), or whether multiple passes through the data are necessary. The last item, *Active*, shows if the approach considers also the problem of exploration.

| **Work** | **Formulation** | **Estimation Tool** | **Representation** | **Data Association** | **Incremental** | **Active** |
|---|---|---|---|---|---|---|
| Standard EKF-SLAM | online SLAM | Kalman Filter | Landmark locations | Yes | Yes | No |
| Lu/Milios [83] | full SLAM | Maximum Likelihood | Point obstacles | Yes | No | No |
| FastSLAM [89] | full SLAM | Particle Filter | Landmark locations | Yes | Yes | No |
| SEIF [125] | online SLAM | Information Filter | Landmark locations | Yes | Yes | No |
| ESDF [41] | trajectory-oriented SLAM | Information Filter | Aligned sensed data | Yes | Yes | No |
| ESEIF[129] | online SLAM | Information Filter | Landmark locations | Yes | Yes | No |
| SLSJF[65] | online SLAM | Information Filter | Landmark locations | Yes | Yes | No |
| SAM[31] | full SLAM | Maximum a Posteriori | Landmark locations | No | No | No |
| Tecnonic SAM[99] | full SLAM | Maximum a Posteriori | Landmark locations | No | Yes | No |
| iSAM[71] | full SLAM | Maximum a Posteriori | Landmark locations | Yes | Yes | No |
| TreeMap [48] | online SLAM | Information Filter | Landmark locations | No | Yes | No |
| TJTF [104] | online SLAM | Information Filter | Landmark locations | No | Yes | No |
| Olson *et al.* [103] | trajectory-oriented SLAM | Maximum Likelihood | Robot Trajectory | No | Yes | No |
| Grisetti *et al.* [53] | trajectory-oriented SLAM | Maximum Likelihood | Robot Trajectory | No | Yes | No |
| Davison *et al.* [27] | online SLAM | Kalman Filter | (Visual) Landmark locations | Yes | Yes | Yes |
| Sim *et al.*[114] | online SLAM | Kalman Filter | Grid Based | Yes | Yes | Yes |
| Feder *et al.* [44] | online SLAM | Kalman Filter | Landmark locations | Yes | Yes | Yes |
| Leung *et al.* [79] | full SLAM | Maximum a Posteriori | Lines | Yes | Yes | Yes |

**Table 2:** Illustrative reviewed works on SLAM

From the literature survey we can conclude that most of the proposed solutions to the SLAM problem are related with the estimation process. However, there are many issues in

SLAM beyond estimation, for instance: long-term SLAM solutions, data association, planning and control (active SLAM), scalability, algorithmic complexity, or the use of multiple robots at the same time.

Much work has to be done to achieve a long-term SLAM. While trajectory-oriented SLAM has many positive characteristics, it has a big problem: its state-space grows unbounded with time, as does the quantity of stored measured data. In addition, in the case where we have collected enough measurements to sufficiently characterize an area, when we return to this previously mapped area, rather than adding more states, we should instead be able to just localize with respect to the finite collection we already have. For long-term SLAM it will become necessary to merge or marginalize out data to bound storage costs and to restrict our representation to environment size (fixed) and not time (unbounded).

Regarding the data association problem, there are issues that are still worth studying. For example, when closing large loops, one needs to evaluate data association tests efficiently. In Visual SLAM, robust place recognition for loop closing as well as robustness to outliers (illumination changes, occlusions, etc.) are necessary.

Moreover, in contrast to the vast number of passive SLAM algorithms, active SLAM has not been studied as extensively. Many works only consider structured environments as well as two-dimensional maps. Although there are works that deal with full 6 DOF motion, they are restricted to small areas. Furthermore, the proposed solutions for exploration in SLAM are generally greedy. Therefore, globally optimal exploration is still and open issue.

Additionally, the information form of the SLAM problem has significant relevance in large-scale mapping, in problems involving many vehicles, and potentially in mixed environments with sensor networks and dynamic landmarks. On the other hand, when dealing with information filters we rely on large linear filters, where linearization errors increase with the size of the environment. An open issue is to keep the advantages of the information form while proposing ways to reduce linearization errors.

# 4 Approach

In the previous section we identified open issues in SLAM. In this section we discuss how we expect to approach some of these open problems.

- Data association to close loops in large environments with delayed-state SLAM:

  A straight forward solution to the loop closing problem is to rely on the pose estimates from the filter and perform data association tests as much and as often as possible. By testing for data association based on the likelihood of estimates, one can avoid some of the problems associated with appearance-based SLAM, such as aliasing for homogeneous or repetitive scenes. However, closing all loops consistent with the likelihood of estimates might add information links that contribute with little information to reduce the estimation error and links that are not consistent with the sensor uncertainties for which the system was trained.

  Our approach to this issue relies on the estimator for the generation of pose constraint hypotheses. Since adding information links for all possible matches produces overconfident estimates that in the long term lead to filter inconsistency, we propose instead a two step loop closure test. First, we check whether two poses are candidates for loop closure with respect to their mean estimates. This can be achieved by testing for the Mahalanobis distance in the same way data association gating is commonly performed during a SLAM update. But instead of adding all these information links, we can limit the candidates to a second test to allow updating using only links with high informative load. In terms of covariances, this happens when a pose with a large covariance can be linked with a pose with a small uncertainty.

- A robot pose marginalization strategy towards a long-term SLAM algorithm:

  We have pointed out that for long-term SLAM, in a trajectory-oriented SLAM approach, it will become necessary to merge or marginalize out data to bound storage costs and to restrict our representation of the environment to a fixed size. We will formalize a pose marginalization strategy based on the information content of each pose. One possibility is to decide which pose to marginalize out based on mutual information scores as well as limiting the effect of the linear approximations added by the links.

  On the other hand, like in feature-based SLAM, if we marginalize out our robot trajectory the information matrix will densify. This suggests that some sort of approximation is required (similar to the pruning strategies employed by feature-based SLAM information filters). As noted in [129], a possibility to control the information matrix sparsification would be to prevent link formation rather than pruning them.

  Our approach to pose marginalization will take into account information theory metrics, such as mutual information, in order to decide when and which poses to marginalize, while taking also into account the computational complexity of performing such strategy.

- Optimal exploration techniques for SLAM as Partially Observable Markov Decision Processes.

  As we have noted, in contrast to the vast number of passive SLAM algorithms, Active SLAM has not been studied as extensively. In this thesis, we seek to propose an exploration estrategy for a delayed-state SLAM based on entropy reduction. While developing the strategy we need to consider that the entropy of the SLAM posterior can be decomposed into the entropy in the pose posterior and into the expected entropy of the map, thus we need to trade off uncertainty in the robot pose (localize the robot) with uncertainty in the map (visit new regions).

- The nonlinearities of perception and action models

  Although the information form of the SLAM problem has significant potential in large-scale mapping, we rely on large linear filters, where linearization errors increase with the size of the environment. A tentative approach to this issue is to reduce the effect of the nonlinearities of motion and perception models either by using an iterative optimization technique or employing an alternative state-space representations.

# 5 Achievements

Here we show the achievements that we have accomplished so far. We present an implementation that employs an Extended Information Filter (EIF), within a delayed-state framework, and a quaternion-based representation, which allows us to avoid gimbal lock singularities. Additionally, we describe the work we have done about informative loop closure detection presented in [67].

The approach presented here constitutes the SLAM estimation backbone for the maps being built for the URUS project [108]. We note that in the scope of this project, relative pose measurements may come in the form of image feature matches or as registration of 3D laser range scans for any of the robot platforms used.

## 5.1 ESDF Implementation with Quaternions

In the delayed-state information-form SLAM we estimate the state vector $\mathbf{x}$, which contains the history of poses from time 0 to $k$, given the history of proprioceptive observations $Z$ and the set of motion commands $U$. Using the canonical parameterization,

$$p(\mathbf{x}|Z,U) = \mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}^{-1}(\mathbf{x};\boldsymbol{\eta},\boldsymbol{\Lambda}), \tag{25}$$

$$\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}, \text{ and } \boldsymbol{\eta} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}. \tag{26}$$

Where $\boldsymbol{\mu}$ is the mean state vector and $\boldsymbol{\Sigma}$ its covariance matrix. $\boldsymbol{\Lambda}$ and $\boldsymbol{\eta}$ are the information matrix and information vector, respectively.

In our implementation one robot pose (the $k$-th component of the state vector $\mathbf{x}$) is defined as follows

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}, \tag{27}$$

where $\mathbf{p} = (x,y,z)^\top$ indicates the position of the robot, and $\mathbf{q}$ is a unit norm quaternion representation of the robot orientation in global coordinates

$$\mathbf{q} = \begin{bmatrix} \mathbf{v} \\ w \end{bmatrix}, \tag{28}$$

with $\mathbf{v} = (q_x, q_y, q_z)^\top$ the imaginary part and $w$ the scalar part of the quaternion.

The noise-free motion model is defined using the compounding operation as defined by Smith *et al.,* [115], and defines the state transition model, relating state components $\mathbf{x}_{k+1}$ and $\mathbf{x}_k$,

$$\begin{aligned} \mathbf{x}_{k+1} &= f(\mathbf{x}_k, \mathbf{u}_k), \\ \mathbf{x}_{k+1} &= \mathbf{x}_k \oplus \mathbf{u}_k, \\ \begin{bmatrix} \mathbf{p}_{k+1} \\ \mathbf{q}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{p}_k + \mathbf{q}_k \otimes \triangle\mathbf{p} \otimes \mathbf{q}_k^{-1} \\ \mathbf{q}_k \otimes \triangle\mathbf{q} \end{bmatrix}, \end{aligned} \tag{29}$$

where $\otimes$ is used to indicate quaternion multiplication, $\mathbf{u}_t$ represents the relative motion given by the odometry data, which is given by the relative travelled distance $\triangle\mathbf{p}$ and the relative rotation change $\triangle\mathbf{q}$.

A first order Taylor series approximation of this model is given by

$$\mathbf{x}_{k+1} \approx f(\boldsymbol{\mu}_k, \mathbf{u}_k) + \mathbf{F}(\mathbf{x}_k - \boldsymbol{\mu}_k) + \mathbf{w}_k, \tag{30}$$

where

$$\mathbf{F} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\boldsymbol{\mu}_k,\mathbf{u}_k}, \tag{31}$$

and zero mean white noise $\mathbf{w}_k$ with covariance $\mathbf{Q}$, used to accomodate for higher order terms and modelling errors.

We form our proprioceptive observation model also using the compounding operations. The noise-free measurement model is given by Equation 32, which tells us how much the robot has moved between any robot pose $\mathbf{x}_i$ and the current pose $\mathbf{x}_k$,

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_i, \mathbf{x}_k) = \ominus\mathbf{x}_i \oplus \mathbf{x}_k, \\ \begin{bmatrix} \mathbf{z}_p \\ \mathbf{z}_q \end{bmatrix} &= \begin{bmatrix} \mathbf{q}_i^{-1} \otimes (\mathbf{p}_k - \mathbf{p}_i) \otimes \mathbf{q}_i \\ \mathbf{q}_i^{-1} \otimes \mathbf{q}_k \end{bmatrix}. \end{aligned} \tag{32}$$

The linearized measurement model is given by

$$\mathbf{z}_k \approx h(\boldsymbol{\mu}_i, \boldsymbol{\mu}_k) + \mathbf{H}(\mathbf{x}_{i,k} - \boldsymbol{\mu}_{i,k}) + \mathbf{v}_k, \tag{33}$$

where $\mathbf{x}_{i,k} = [\mathbf{x}_i^\top, \mathbf{x}_k^\top]^\top$ , $\mathbf{v}_k$ is zero mean white measurement noise with covariance $\mathbf{R}$, and

$$\mathbf{H} = \begin{bmatrix} \left.\frac{\partial h}{\partial \mathbf{x}_i}\right|_{\boldsymbol{\mu}_i} & \left.\frac{\partial h}{\partial \mathbf{x}_k}\right|_{\boldsymbol{\mu}_k} \end{bmatrix}. \tag{34}$$

In the delayed-state framework we do not maginalize out past robot poses as in other classical SLAM approaches such as the EKF and the EIF. Instead, we append the time-propagated robot pose $\mathbf{x}_{t+1}$ to the state vector, obtaining the prior probability distribution

$$p(\mathbf{x}_{0:k}, \mathbf{x}_{k+1}|Z^k, U^{k+1}) = p(\mathbf{x}_{0:k}|Z^k, U^k)\, p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_{k+1}), \tag{35}$$

where $\mathbf{x}_{0:k}$ represent the robot trajectory before time $k+1$, $Z^k$ and $U^k$ are the history of observations and odometry increments up to time $k$, respectively. This probability is factored into the product of the state posterior at time $k$ and the transition probability multiplied by the prior probability —i.e. the posterior distribution computed at time $k$. For Gaussian distributions, the parameters $\boldsymbol{\eta}$ and $\boldsymbol{\Lambda}$ of Equation 35 in the form of 25 are given by

$$\boldsymbol{\eta}_{k,k+1} = \bar{\boldsymbol{\eta}}_{k,k+1} + \mathbf{F}_{\text{aug}}^\top \mathbf{Q}^+ (f(\boldsymbol{\mu}_k, \mathbf{u}_k) - \mathbf{F}\boldsymbol{\mu}_k) \tag{36}$$

and

$$\Lambda_{k:k+1,k:k+1} = \bar{\Lambda}_{k:k+1,k:k+1} + \mathbf{F}_{\text{aug}}^\top \mathbf{Q}^+ \mathbf{F}_{\text{aug}}, \tag{37}$$

in which

$$\mathbf{F}_{\text{aug}} = \begin{bmatrix} -\mathbf{F} & \mathbf{I} \end{bmatrix}, \tag{38}$$

and $\bar{\boldsymbol{\eta}}_{k+1}$ and $\bar{\boldsymbol{\Lambda}}_{k+1}$ represent the posterior information vector and information matrix at time $k$, with zero entries for time $k+1$, representing infinite uncertainty for that robot pose.

The augmentation process introduces information only between the new robot pose $\mathbf{x}_{k+1}$ and the previous one $\mathbf{x}_k$. Moreover, the shared information between the new pose $\mathbf{x}_{k+1}$ and the rest of the robot trajectory $\mathbf{x}_{0:k-1}$ is always zero when we have not closed any loop. This matrix results in a naturally sparse information matrix with a tridiagonal block structure.

In this implementation, we model rotation noise in terms of Euler angles, and use a linear propagation of the white noise $\mathbf{w}_e \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_e)$ that expresses rotation error in Euler angles, to compute the noise $\mathbf{w}_q$, expressed in quaternions. For that purpose we define the function

$$g : \mathcal{X}_e \to \mathcal{X}_q, \tag{39}$$

which transforms robot orientations in Euler angles $\mathcal{X}_e \subseteq \mathrm{SO}(3)$ to robot orientations in quaternions $\mathcal{X}_q$. Linearizing $g$ about the mean robot orientation expressed in Euler angles $\boldsymbol{\mu}_e$

$$g(\mathbf{x}_e) \approx g(\boldsymbol{\mu}_e) + \mathbf{G}(\mathbf{x}_e - \boldsymbol{\mu}_e), \tag{40}$$

where

$$\mathbf{G} = \left. \frac{\partial g}{\partial \mathbf{x}_e} \right|_{\boldsymbol{\mu}_e} . \tag{41}$$

$\mathbf{Q}^+$ is the pseudoinverse of the motion noise covariance expressed in quaternions. To obtain it, we first transform uncertainty in Euler angles to quaternionrs with

$$\mathbf{Q}_q = \mathbf{G}\mathbf{Q}_e\mathbf{G}^\top, \tag{42}$$

which is rank deficient by one because of the normalized quaternion representation. Its pseudoinverse is computed with

$$\mathbf{Q}_q^+ = \mathbf{G}(\mathbf{G}^\top\mathbf{G}\mathbf{Q}_e\mathbf{G}^\top\mathbf{G})^{-1}\mathbf{G}^\top, \tag{43}$$

and finally,

$$\mathbf{Q}^+ = \begin{bmatrix} \mathbf{Q}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_q^+ \end{bmatrix}, \tag{44}$$

where $\mathbf{Q}_p$ represents the translational components of the noise covariance $\mathbf{Q}$.

After augmenting the state we add observations with the EIF update equations,

$$\boldsymbol{\eta}_{i,k+1} = \bar{\boldsymbol{\eta}}_{i,k+1} + \mathbf{H}^\top\mathbf{R}^+ \left(\mathbf{z}_{k+1} - h(\boldsymbol{\mu}_i, \bar{\boldsymbol{\mu}}_{k+1}) + \mathbf{H}\bar{\boldsymbol{\mu}}_{i,k+1}\right), \tag{45}$$

$$\boldsymbol{\Lambda}_{i:k+1,i:k+1} = \bar{\boldsymbol{\Lambda}}_{i:k+1,i:k+1} + \mathbf{H}^\top\mathbf{R}^+\mathbf{H}, \tag{46}$$

where $\mathbf{z}_{k+1}$ is the observation at time $k+1$. Here again $\mathbf{R}^+$ is the pseudoinverse of the covariance noise, in this case, from the observation model, which is given by

$$\mathbf{R}^+ = \begin{bmatrix} \mathbf{R}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{G}(\mathbf{G}^\top\mathbf{G}\mathbf{R}_e\mathbf{G}^\top\mathbf{G})^{-1}\mathbf{G}^\top \end{bmatrix}. \tag{47}$$

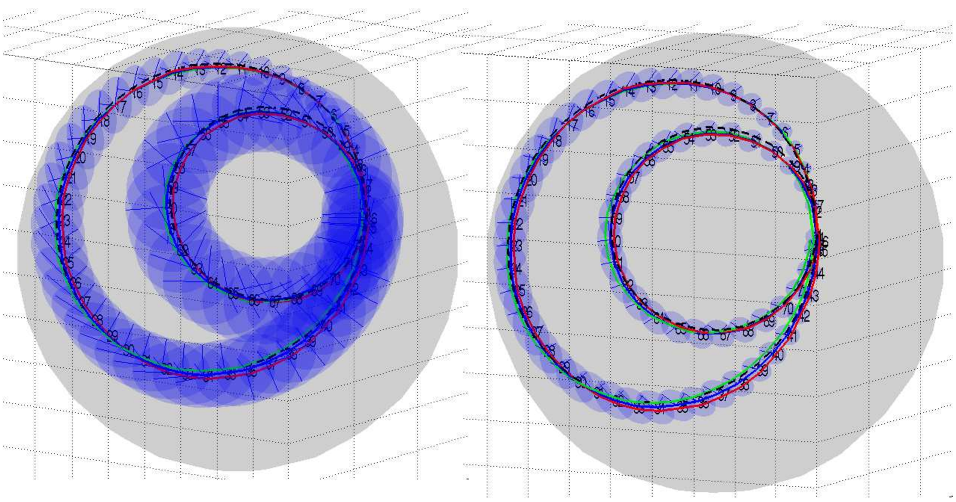


**Figure 2:** Simulated robot trajectory on a sphere. The left figure shows the estimated trajectory before loop closure, whereas the right image shows the estimated trajectory after the loop is closed.

In the same way as with the prediction step, given the two-block size of the measurement Jacobian $\mathbf{H}$ (Equation 34) [125], only the four blocks relating poses $i$ and $k+1$ in the information matrix will be updated.

Note also that at each iteration, quaternion normalization must be enforced, i.e.,

$$g_n(\mathbf{q}) = \frac{\mathbf{q}}{\parallel \mathbf{q} \parallel}, \tag{48}$$

$$\mathbf{G}_n = \left.\frac{\partial g_n}{\partial \mathbf{q}}\right|_{\boldsymbol{\mu}_q}, \tag{49}$$

substituting the state mean and covariance with

$$\hat{\boldsymbol{\mu}}_q = g_n(\boldsymbol{\mu}_q), \tag{50}$$

and

$$\hat{\boldsymbol{\Sigma}}_q = \mathbf{G}_n \boldsymbol{\Sigma}_q \mathbf{G}_n^\top. \tag{51}$$

The filter has been implemented and simulated for a robot moving about a sphere following a "Hawaiin necklace" manoeuvre with a unicycle motion model [61]. Figure 2 shows the simulated trajectory before and after loop closure. These simulations verified that our quaternion-based SLAM implementation works well for full 3D motion.

## 5.2 Informative Loop Closure Detection

Two phases can be distinguished during the loop-closing process. First, we need to detect the possibility of a loop closure event and then, we must certify the presence of such loop closure from visual data. The likelihood of pose estimates are valuable in detecting possible loop closures. A comparison of the current pose estimate with the history of poses can tell whether the robot is in the vicinity of a previously visited place, in terms of both the global position and the orientation. This is achieved by measuring the Mahalanobis distance from the prior estimate to all previously visited locations, i.e., for all $0 < i < k$,

$$d_M^2 = (\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_i)^\top \left( \frac{\boldsymbol{\Sigma}_{k+1} + \boldsymbol{\Sigma}_i}{2} \right)^{-1} (\boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_i). \tag{52}$$

An exact computation of $\Sigma_{k+1}$ and $\Sigma_i$ requires the inverse of $\bar{\Lambda}$, which can be computed in linear time using conjugate gradient techniques [41]. Motivated by [125], these covariances can be efficiently approximated in constant time from their Markov blankets. Note also that Equation 52 does not take into account the cross correlation between poses in the Mahalanobis metric, but this can be done with no substantial extra effort. The only difference is that instead of computing individual Markov blankets for each pose, the combined Markov blanket is used.

The average covariance is used to accommodate for the varying levels of estimation uncertainty both on the pose prior being evaluated, and on the past pose being compared. In case of a normal distribution, the Mahalanobis distance follows the $\chi^2$-square distribution with n-1 degrees of freedom.

Many nearby poses will satisfy this condition, as shown in Figure 3b. At the start of a SLAM run, when covariances are small, only links connecting very close poses will satisfy the test. But, as error accumulates, pose covariances grow covering larger and larger areas of matching candidates.

For long straight trajectories having corresponding visual features, information links could even be established. Some aid in reducing the effect of large pose constraints, because orientation variance is usually larger than position variance, and pose covariance ellipsoids usually align orthogonal with respect to the direction of motion.

Due to linearization effects, adding information links for all possible matches produces overconfident estimates that in the long run lead to filter inconsistency. Thus, our update

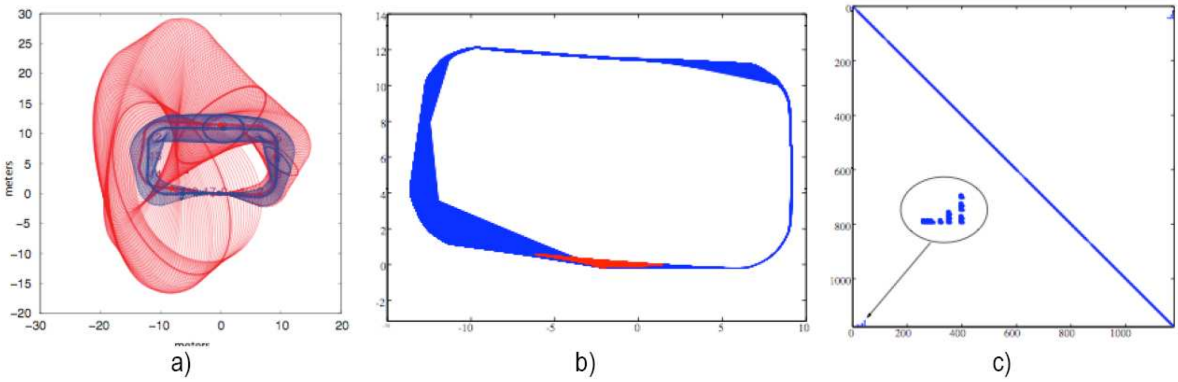


**Figure 3:** A mobile robot has performed a loop trajectory. a) Prior to adding the information relevant to the loop, a match hypothesis test must be confirmed. If asserted, we could change the overall uncertainty in the trajectory from the red hyperellipsoids to the ones in blue. b) If the test is made using conventional statistical tools, such as the Mahalanobis test, all the possible data association links indicated in blue should be verified. With a test based in information content just the links indicated in red should be verified. c) Sparse information matrix using a "delayed-state" SLAM representation. A loop closure event adds only a few non-zero off-diagonal elements to the matrix (see zoomed region).

procedure must pass a second test. The aim of this second test is to allow updating using only links with high informative load. In terms of covariances, this happens when a pose with a large covariance can be linked with a pose with a small uncertainty.

$$d_B = \frac{1}{2} ln \frac{\left| \frac{\mathbf{\Sigma}_{k+1} + \mathbf{\Sigma}_i}{2} \right|}{\sqrt{\mid \mathbf{\Sigma}_{k+1} \mid\mid \mathbf{\Sigma}_i \mid}} \tag{53}$$

The above expression refers to the second term of the Bhattacharyya distance, and gives a measure of separability in terms of covariance difference [50]. This test is typically used to discern between to distinct classes with close means but varying covariances. We can see however that it also can be used to fuse two observations of the same event with varying covariance estimates. Given that, the value of $d_B$ increases as the two covariances $\mathbf{\Sigma}_{k+1}$ and $\mathbf{\Sigma}_i$ are more different. The Bhattacharyya covariance separability measure is symmetric, and we need to test whether the current pose covariance is larger than the $i$-th pose it is being compared with. This is done by analyzing the area of uncertainty of each estimate by comparing the determinants of $\mid \mathbf{\Sigma}_{k+1} \mid$ and $\mid \mathbf{\Sigma}_i \mid$. The reason is that we only want to update the overall estimate with information links to states that had smaller uncertainty than the current state. Figure 3b shows in red the remaining links after the second test.

In a second phase we still must certify the presence of a loop closure event to update the entire pose estimate and to reduce overall uncertainty. When an image correspondence can be established, the computed pose constraint is used in a one-step update of the information filter, as shown in Equations 46 and 45. A one-step update in information form changes the entire history of poses adding a linear number of non-zero off-diagonal elements in the information matrix as shown in the Figure 3c. The sparsity can be controlled by reducing the confidence on image registration when testing for loop-closure event.

# 6 Work Plan and Calendar

In this section we present the tasks required to develop this Thesis. Part of our previous work is also contained in these tasks.

- **Task 1 (T1):** Courses
  **Term:** 9 months
  Courses of the PhD in Vision, Control and Robotics and complementary courses (summer schools)

- **Task 2 (T2):** State-of-the-Art review
  **Term:** 18 months
  In-depth study of SLAM approaches, mainly focusing on the works dealing with scalability issues, computational complexity, data association, Visual SLAM, nonlinearities in perception and action models, and Active SLAM. This task includes the thesis proposal preparation.

- **Task 3 (T3):** Analysis and implementation of loop closure techniques
  **Term:** 12 months
  Study of different metrics for data association hypothesis testing in a delayed-state SLAM framework and experiments to acquire an image data base with the SEGWAY RMP platform. Report the proposed solution and results.

- **Task 4 (T4):** 6 DOF SLAM implementation with reduced linearization errors
  **Term:** 10 months
  Implementation of a trajectory-oriented SLAM approach in 6 DOF, selecting an appropriate rotation representation to avoid singularities. Reduce the effect of the nonlinearities of motion and perception models. Report the proposed solution and results.

- **Task 5 (T5):**Pose Marginalization
  **Term:** 10 months
  Formalization of a pose marginalization strategy to achieve scalability and reduced storage costs in a trajectory-oriented SLAM approach. Validation of the proposed approach by simulations and real data. Report the proposed solution and results.

- **Task 6 (T6):** Propose exploration strategies
  **Term:** 12 months
  Formalization of exploration strategies. Evaluation by means of simulations. Report the proposed solution and results.

- **Task 7 (T7):** Propose a unification approach of SLAM and exploration techniques.
  **Term:** 8 months
  Combine all the previously developed methods to achieve SLAM exploration. Validation of the proposed approach by simulations and experiments with a robotic platform. Report the proposed solution and results.

- **Task 8 (T8):** Thesis report and thesis defense
  **Term:** 6 months
  In this task all the reports will be integrated to complete the Thesis report writing. This task also includes the preparation of the Thesis defense and preparations of international journal papers.

- **Task 9 (T9):** URUS experiments
  **Term:** 4 months
  Since this Thesis is part of the URUS project [108], besides the necessary experiments to evaluate the proposed techniques we need to perform additional experiments in order to incorporate the results of this Thesis into the project.

- **Task 10 (T10):** Research Stay
  **Term:** 6 months
  It is expected to do one or two short stays in another University to work in the Thesis research topic. They are yet to be defined. Candidate collaborations include ACFR (Australia), LAAS (France), UniZar (Spain), and Georgia Tech (USA).

Figure 4 shows an orientative planning of the thesis work, following the tasks mentioned above.

## 6.1 Previous Work and Contributions

- Courses of the PhD in Vision, Control and Robotics (September 2006 - June 2007) **(T1)**

  *Control y Programación de Robots, Robotización Avanzada, Control de Robots, Planificación de Trayectorias y Detección de Colisiones en Robotica, Visión por Ordenador (I): Segmentación, Modelado y Reconocimiento,* and *Visión por Ordenador (II): Procesamiento y Segmentación de la Imagen.*

- Summer schools **(T1)**

  Player Summer School on Cognitive Robotics (PSSCR07) . Study of topics on Cognitive Robotics and applications of the Player/Stage middleware. Technische Universität München (TUM, Technical University of Munich). Munich, August 13th - 27th, 2007.

  Summer School on Image and Robotics (SSIR08). Study of Computer Vision and Robotics topics. Université Blaise-Pascal (Blaise Pascal University). Clermont-Ferrand, June 22nd - July 4th, 2008.

- State-of-the-Art review **(T2)**

  Study of the different techniques applied in the literature to achieve our objective and development of the thesis proposal.

- Loop closure techniques **(T3)**

  During the first year of my doctoral studies I have contributed with a novel test for data association hypothesis testing within a Delayed-State SLAM framework [67]. This research was directed by Dr. V. Ila, Dr. J. Andrade-Cetto, and Prof. A. Sanfeliu. This work was presented in the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, in San Diego, CA, USA.

- Implementation of a 6 DOF SLAM **(T4)**

  Implementation of an Extended Information Filter (EIF), within a delayed-state framework, and a quaternion-based representation. The implemented method constitutes the SLAM estimation backbone for the maps being built for the URUS project [108].

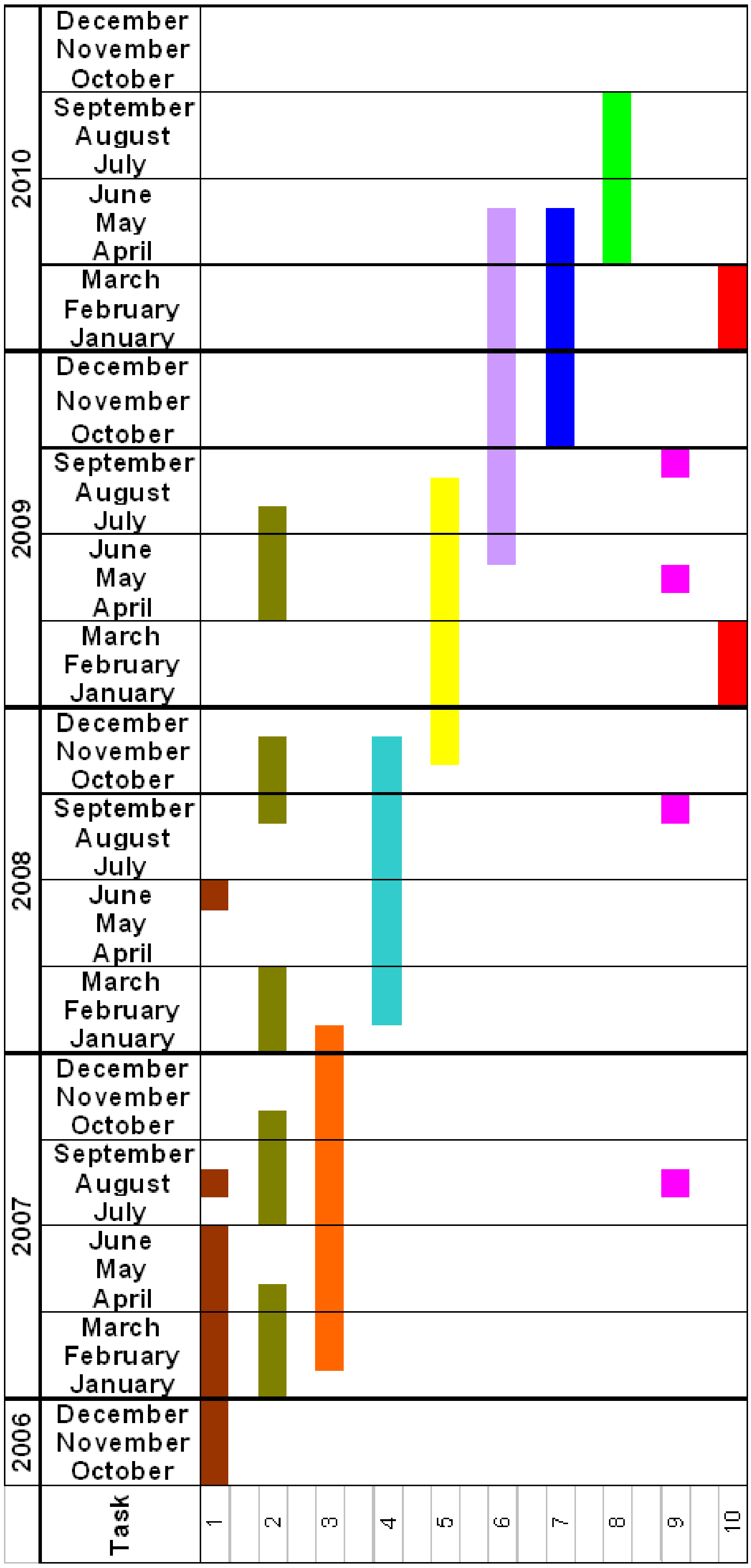



**Figure 4:** Work Plan

## References


[1] M. Agrawal, K. Konolige, and M. R. Blas. CenSurE: center surround extremas for realtime feature detection and matching. In *European Conf. Comput. Vision*, Marseille, 2008. To appear.

[2] J. Andrade-Cetto and A. Sanfeliu. Temporal landmark validation in CML. In *IEEE Int. Conf. Robot. Automat.*, pages 1576–1581, Taipei, 2003.

[3] J. Andrade-Cetto and A. Sanfeliu. The effects of partial observability when building fully correlated maps. *IEEE Trans. Robot.*, 21(4):771–777, 2005.

[4] J. Andrade-Cetto, T. Vidal-Calleja, and A. Sanfeliu. Unscented transformation of vehicle states in SLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 324–329, Barcelona, 2005.

[5] S. Argamon-Engelson. Using image signatures for place recognition. *Pattern Recogn. Lett.*, 19(4), 1998.

[6] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments.* PhD thesis, The University of Sydney, Australian Center for Field Robotics, Sydney, 2002.

[7] T. Bailey, J. Nieto, and E. Nebot. Consistency of the FastSLAM algorithm. In *IEEE Int. Conf. Robot. Automat.*, pages 424–429, Orlando, May 2006.

[8] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, New York, 2001.

[9] A. Bennett and J. J. Leonard. A behavior-based approach to adaptive feature mapping with autonomous underwater vehicles. *IEEE J. of Oceanic Engineering*, 25(2):213–226, 2000.

[10] D. Blackwell. Conditional expectation and unbiased sequential estimation. *Ann.Math.Statist*, 18(1):105–110, 1947.

[11] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas framework for scalable mapping. In *IEEE Int. Conf. Robot. Automat.*, pages 1899–1906, Taipei, 2003.

[12] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the *atlas* framework. *Int. J. Robot. Res.*, 23(12):1113–1139, 2004.

[13] M. Bosse, R. Rikoski, J. Leonard, and S. Teller. Vanishing points and 3D lines from omnidirectional video. In *IEEE Int. Conf. Image Process.*, pages 513–516, Rochester, 2002.

[14] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 540–545, Lausanne, 2002.

[15] M. Bryson and S. Sukkarieh. An information-theoretic approach to autonomous navigation and guidance of an uninhabited aerial vehicle in unknown environments. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 3770–3775, Edmonton, 2005.

[16] J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardós. The SPMap: A probabilistic framework for simultaneous localization and map building. *IEEE Trans. Robot. Automat.*, 15(5):948–952, October 1999.

[17] J. A. Castellanos, J. Neira, and J. D. Tardós. Multisensor fusion for simultaneous localization and map building. *IEEE Trans. Robot. Automat.*, 17(6):908–914, December 2001.

[18] J.A. Castellanos, J.M. Martínez, J. Neira, and J. D. Tardós. Experiments in multisensor mobile robot localization and map building. In *3rd IFAC Symp. on Intelligent Autonomous Vehicles*, pages 173–178, Madrid, 1998.

[19] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. "MFm": 3-d motion from 2-d motion causally integrated over time. In *European Conf. Comput. Vision*, pages 734–750, Dublin, 2000.

[20] K. S. Chong and L. Kleeman. Mobile robot map building for an advanced sonar array and accurate odometry. *Int. J. Robot. Res.*, 18(1):20–36, 1999.

[21] J. Civera, J. A. Davison, and J. M. M. Montiel. Interacting multiple model monocular SLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 3704–3709, Pasadena, May 2008.

[22] D.M. Cole and P.M. Newman. 3D SLAM in outdoor environments. In *IEEE Int. Conf. Robot. Automat.*, pages 1556–1563, Orlando, May 2006.

[23] I. J. Cox and J. J. Leonard. Modeling a dynamic environment using a bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344, 1994.

[24] M. Cummins and P. Newman. Accelerated appearance-only SLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 1828–1833, Pasadena, May 2008.

[25] A. J. Davison. *Mobile Robot Navigation Using Active Vision*. PhD thesis, Univ. Of Oxford, 1998.

[26] A. J. Davison and N. Kita. 3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain. In *IEEE Conf. Comput. Vision Pattern Recog.*, pages 384–391, Kauai, 2001.

[27] A. J. Davison and D. W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(7):865–880, 2002.

[28] A. J. Davison and D.W. Murray. Mobile robot localisation using active vision. In *European Conf. Comput. Vision*, pages 809–825, Freiburg, 1998.

[29] A. J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(6):1052–1067, 2007.

[30] F. Dellaert. Square root sam: Simultaneous location and mapping via square root information smoothing. In *Robotics Science and Systems Conf.*, Cambridge, June 2005.

[31] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Int. J. Robot. Res.*, 25(12):1181–1204, December 2006.

[32] G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Automat.*, 17(3):229–241, 2001.

[33] 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. Trevelya, editors, *Experimental Robotics VI*, volume 250/2000 of *Lecture Notes in Control and Information Sciences*, pages 265–274. Springer Berlin Heidelberg, London, 2000.

[34] T. Duckett, S. Marsland, and J. Shapiro. Learning globally consistent maps by relaxation. In *IEEE Int. Conf. Robot. Automat.*, pages 3841–3846, San Francisco, 2000.

[35] H. Durrant-Whyte, S. Majumder, M. de Battista, and S. Scheding. A Bayesian algorithm for simultaneous localisation and map building. In *10th International Symposium on Robotics Research*, Lorne, November 2001.

[36] H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE J. Robot. Automat.*, 4(1):23–31, February 1988.

[37] E. Eade and T. Drummond. Scalable monocular slam. In *IEEE Conf. Comput. Vision Pattern Recog.*, pages 469–476, New York, 2006.

[38] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Int. Joint Conf. Artificial Intell.*, pages 1135–1142, Acapulco, 2003.

[39] A. Eliazar and R. Parr. DP-SLAM 2.0. In *IEEE Int. Conf. Robot. Automat.*, pages 1314–1320, New Orleans, 2004.

[40] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. *IEEE Trans. Robot.*, 21(4):588–596, August 2005.

[41] R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robot.*, 22(6):1100–1114, December 2006.

[42] R. M. Eustice, H. Singh, J. J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS titanic with SLAM information filters. In *Robotics Science and Systems Conf.*, pages 57–64, Cambridge, 2005.

[43] R. M. Eustice, M. Walter, and J. J. Leonard. Sparse extended information filters: insights into sparsification. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 3281–3288, Edmonton, 2005.

[44] H.J.S. Feder, J. J. Leonard, and C.M. Smith. Adaptive mobile robot navigation and mapping. *Int. J. Robot. Res.*, 18:650–668, 1999.

[45] J. Folkesson and H. I. Christensen. Graphical SLAM –a self-correcting map. In *IEEE Int. Conf. Robot. Automat.*, pages 791–798, New Orleans, 2004.

[46] E. Foxlin. Generalized architecture for simultaneous localization, auto-calibration and map-building. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 527–533, Lausanne, 2002.

[47] U. Frese. A proof for the approximate sparsity of slam information matrices. In *IEEE Int. Conf. Robot. Automat.*, pages 329–335, Barcelona, 2005.

[48] U. Frese. Treemap: An $o(logn)$ algorithm for indoor simultaneous localization and mapping. *Auton. Robot.*, 21(2):103–122, September 2006.

[49] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Trans. Robot.*, 21(2):1–12, 2005.

[50] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, 2nd edition, 1990.

[51] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3d. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 3472–3478, San Diego, 2007.

[52] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson, and W. Burgard. Online constraint network optimization for efficient maximum likelihood mapping. In *IEEE Int. Conf. Robot. Automat.*, pages 1880–1885, Pasadena, May 2008.

[53] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Robotics Science and Systems Conf.*, Atlanta, 2007.

[54] J. E. Guivant and E. M. Nebot. Optimization of simultaneous localization and map-builidng algorithm for real-time implementation. *IEEE Trans. Robot. Automat.*, 17(3):242–257, 2001.

[55] J. E. Guivant and E. M. Nebot. Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. *IEEE Trans. Robot. Automat.*, 19(4):749–755, 2003.

[56] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE Int. Symp. on Comput. Intell. in Robot. and Automat.*, pages 318–325, Monterey, 1999.

[57] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, volume 1, pages 206–211, Las Vegas, 2003.

[58] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3d a models of indoor and outdoor environments with a mobile robot. In *The Fourth European workshop on advanced mobile robots EUROBOT'01*, pages 91–98, 2001.

[59] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in SLAM. In *Int. Sym. Robot. Res.*, Siena, 2003.

[60] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, 2000.

[61] M. P. Hennessey. Visualizing the motion of a unicycle on a sphere. *Int. J. Model. Simul.*, 26(1):69–79, 2006.

[62] S. Holmes, G. Klein, and D. Murray. A square root unscented kalman filter for Visual Monoslam,. In *IEEE Int. Conf. Robot. Automat.*, pages 3710–3716, Pasadena, May 2008.

[63] J. Horn and G. Schmidt. Continuous localization of a mobile robot based on 3D-laser-range-data, predicted sensor images, and dead-reckoning. *Robot. Auton. Syst.*, pages 99–118, 2005.

[64] A. Howard, M. Mataric, and G. Sukhatme. Relaxation on a mesh: a formalism for generalized localization. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, volume 2, pages 1055–1060, Maui, 2001.

[65] S. Huang, Z. Wang, and G. Dissanayake. Exact state and covariance sub-matrix recovery for submap based sparse EIF SLAM algorithm. In *IEEE Int. Conf. Robot. Automat.*, pages 1868–1873, Pasadena, May 2008.

[66] V. Ila, J. Andrade-Cetto, and A. Sanfeliu. Outdoor delayed-state visually augmented odometry. In *6th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Toulouse, 2007.

[67] V. Ila, J. Andrade-Cetto, R. Valencia, and A. Sanfeliu. Vision-based loop closing for delayed state robot mapping. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 3892–3897, San Diego, 2007.

[68] J. D. Tardòs J. Neira and J.A. Castellanos. Linear time vehicle relocation in SLAM. In *IEEE Int. Conf. Robot. Automat.*, volume 1, pages 427–423, Taipei, 2003.

[69] S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *SPIE Int. Sym. Aerospace/Defense Sensing, Simulation, Controls*, pages 182–193, Orlando, 1997.

[70] I. Jung and S. Lacroix. High resolution terrain mapping using low altitude aerial stereo imagery. In *IEEE Int. Conf. Comput. Vision*, pages 946–951, Nice, 2003.

[71] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast incremental smoothing and mapping with efficient data association. In *IEEE Int. Conf. Robot. Automat.*, pages 1670–1677, Rome, 2007.

[72] N. Karlsson, E.D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M.E. Munich. The vSLAM algorithm for robust localization and mapping. In *IEEE Int. Conf. Robot. Automat.*, pages 24–29, Barcelona, 2005.

[73] J.H. Kim and S. Sukkarieh. Airborne simultaneous localization and map building. In *IEEE Int. Conf. Robot. Automat.*, pages 406–411, Taipei, 2003.

[74] J. Knight, A. Davision, and I. Reid. Towards constant SLAM using postponement. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, volume 1, pages 405–413, Maui, 2001.

[75] P. Kohlhepp, P. Pozzo, M. Walter, and R. Dillmann. Sequential 3d-slam for mobile action planning. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 722–729, Sendei, 2004.

[76] K. Konolige. Large-scale map-making. In *National Conf. on Artificial Intelligence*, pages 457–463, San Jose, 2004.

[77] K. Konolige, M. Agrawal, and J. Sola. Large scale visual odometry for rough terrain. In *Int. Sym. Robot. Res.*, Hiroshima, 2007.

[78] J. J. Leonard and H. J. S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. *Proceedings of the Ninth International Symposium on Robotics Research, Salt Lake City, Utah*, 1999.

[79] C. Leung, S. Huang, and G. Dissanayake. Active SLAM for structured environments. In *IEEE Int. Conf. Robot. Automat.*, pages 1898–1903, Pasadena, May 2008.

[80] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to learn 3D models of indoor environments with mobile robots. In *Int. Conf. Machine Learning*, pages 329–336, Massachusetts, June 2001.

[81] Y. Liu and S. Thrun. Results for outdoor-SLAM using sparse extended information filters. In *IEEE Int. Conf. Robot. Automat.*, volume 1, pages 1227–1233, Taipei, 2003.

[82] D.G. Lowe. Object recognition from local scale-invariant features. In *IEEE Int. Conf. Comput. Vision*, pages 1150–1157, Corfu, 1999.

[83] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Auton. Robot.*, 4(4):333–349, 1997.

[84] A. A. Makarenko, S. B. Williams, H. F. Bourgoult, and H. F. Durrant-Whyte. An experiment in integrated exploration. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 534–539, Lausanne, 2002.

[85] T. K. Marks, A. Howarda, M. Bajracharya, G.W. Cottrell, and L. Matthies. Gamma-SLAM: using stereo vision and variance grid maps for slam in unstructured environments. In *IEEE Int. Conf. Robot. Automat.*, pages 3717–3724, Pasadena, May 2008.

[86] R. Martinez-Cantin and J. A. Castellanos. Unscented SLAM for large-scale outdoor environments. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 328–333, Edmonton, 2005.

[87] P. F. McLauchlan and D.W. Murray. A unifying framework for structure and motion recovery from image sequences. In *IEEE Int. Conf. Comput. Vision*, pages 314–320, Massachusetts, 1995.

[88] N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

[89] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. In *18th National Conf. on Artificial Intelligence*, pages 593–598, Edmonton, 2002.

[90] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-SLAM 2.0: An improved particle filtering for simultaneous localization an mapping that provably converges. In *Int. Joint Conf. Artificial Intell.*, pages 1151–1156, Acapulco, 2003.

[91] J. M. M. Montiel, J. Civera, and A. J. Davison. Unified inverse depth parametrization for monocular SLAM. In *Robotics Science and Systems Conf.*, Philadelphia, 2006.

[92] H. P. Moravec. *Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids.* CMU technical report CMU-RI-TR-96-34, 1996.

[93] P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *Int. Sym. Experimental Robotics*, pages 327–346, Montreal, 1989.

[94] K. Murphy and S.Russell. Rao-blackwellized particle filtering for dynamic bayesian networks. In A. Doucet, N. Freitas, and N.Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 499–515. Springer-Verlag, Lorne, Australia, 2001.

[95] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robot. Automat.*, 17(6):890–897, 2001.

[96] P. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem.* PhD thesis, The University of Sydney, Sydney, 1999.

[97] P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *IEEE Int. Conf. Robot. Automat.*, pages 1556–1563, Orlando, May 2006.

[98] P. M. Newman, M. Bosse, and J. J. Leonard. Autonomous feature-based exploration. In *IEEE Int. Conf. Robot. Automat.*, pages 1234–1240, Taipei, 2003.

[99] K. Ni, D. Steedly, and F. Dellaert. Tectonic SAM: Exact, out-of-core, submap-based SLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 1678–1685, Rome, 2007.

[100] J. Nieto, J. Guivant, and E. Nebot. The hybrid metric maps (HYMMs): a novel map representation for DenseSLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 391–396, New Orleans, 2004.

[101] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Conf. Comput. Vision Pattern Recog.*, pages 652–659, Washington, 2004.

[102] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6D slam with application in autonomous mine mapping. In *IEEE Int. Conf. Robot. Automat.*, pages 1998–2003, New Orleans, April 2004.

[103] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *IEEE Int. Conf. Robot. Automat.*, pages 2262–2269, Orlando, May 2006.

[104] M. A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Int. Joint Conf. Artificial Intell.*, pages 1157–1164, Acapulco, 2003.

[105] C. R. Rao. Information and accuracy obtainable in estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37:81–91, 1945.

[106] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation–mobile robot navigation with uncertainty in dynamic environments. In *IEEE Int. Conf. Robot. Automat.*, pages 35–40, Detroit, May 1999.

[107] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *IEEE Int. Conf. Comput. Vision*, page 59, Bombay, 1998.

[108] A. Sanfeliu and J. Andrade-Cetto. Ubiquitous networking robotics in urban settings. In *IEEE/RSJ IROS Workshop on Network Robot Systems*, pages 14–18, Beijing, 2006.

[109] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *Int. J. Robot. Res.*, 21(8):735–758, 2002.

[110] V. Sequeira, J. G. M. Goncalves, and M.I.Ribeiro. 3D reconstruction of indoor environments. In *IEEE Int. Conf. Image Process.*, volume 2, Lausanne, September 1996.

[111] R. Sim. Stabilizing information-driven exploration for bearings-only SLAM using range gating. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 3396–3401, Edmonton, 2005.

[112] R. Sim. Stable exploration for bearings-only SLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 2422–2427, Barcelona, 2005.

[113] R. Sim, P. Elinas, and J. J. Little. A study of the rao-blackwellised particle filter for efficient and accurate Vision-Based SLAM. *Int. J. Comput. Vision*, 74(3):303–318, 2007.

[114] R. Sim and N. Roy. Global A-optimal robot exploration in SLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 673–678, Barcelona, 2005.

[115] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 1:167–193, 1990.

[116] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[117] J. Sola, A. Monin, M. Devy, and T. Lemaire. Undelayed initialization in bearing only SLAM. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 2499–2504, Edmonton, 2005.

[118] C. Stachniss and W. Burgard. Exploring unknown environments with mobile robots using coverage maps. In *Int. Joint Conf. Artificial Intell.*, pages 1127–1134, Acapulco, 2003.

[119] H. Surmann, A. Nuchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robot. Auton. Syst.*, 45(3):181–198, December 2003.

[120] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard. Robust mapping and localization in indoor environments using sonar data. *Int. J. Robot. Res.*, 21(4):311–330, 2002.

[121] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*, World Scientific Series in Robotics and Intelligent Systems. Morgan Kaufmann, 2002.

[122] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3Dmapping. In *IEEE Int. Conf. Robot. Automat.*, pages 321–328, San Francisco, 2000.

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

[124] S. Thrun and J. J. Leonard. Simultaneous localization and mapping. In S. Bruno and K. Oussama, editors, *Springer Handbook of Robotics*, volume E, chapter 37, pages 871–889. Springer Berlin Heidelberg, 2008.

[125] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Int. J. Robot. Res.*, 23(7-8):693–716, July 2004.

[126] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *IEEE Int. Conf. Robot. Automat.*, pages 1023–1029, San Francisco, 2000.

[127] T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D.W. Murray. Active control for single camera SLAM. In *IEEE Int. Conf. Robot. Automat.*, pages 1930–1936, Orlando, May 2006.

[128] M. Walter, R. Eustice, and J. Leonard. A provably consistent method for imposing exact sparsity in feature-based SLAM information filters. In *Int. Sym. Robot. Res.*, pages 214–234, San Francisco, CA, October 2005.

[129] M.R Walter, R. M. Eustice, and J. J. Leonard. Exactly sparse extended information filters for feature-based SLAM. *Int. J. Robot. Res.*, 26(4):335–359, April 2007.

[130] Z. Wang, S. Huang, and G. Dissanayake. D-SLAM: A decoupled solution to simultaneous localization and mapping. *Int. J. Robot. Res.*, 26(2):187–204, February 2007.

[131] Z. Wang, S. Huang, and G. Dissanayake. Tradeoffs in slam with sparse information filters. In *6th International Conference on Field and Service Robotics*, pages 9–12, Chamonix, 2007.

[132] J. Weingarten and R. Siegwart. EKF-based 3D SLAM for structured environment reconstruction. In *IEEE/RSJ Int.Conf.on Intelligent Robots and Systems*, pages 3834–3839, Edmonton, 2005.

[133] S. B. Williams. *Efficient solutions to autonomous mapping and navigation problems*. PhD thesis, The University of Sydney, Australian Center for Field Robotics, Sydney, 2001.

[134] B. Yamauchi. Frontier-based exploration using multiple robots. In *Int. Conf. Autonomous Agents*, pages 47–53, Minneapolis, 1998.