The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Towards the Exploitation of Prior Information in SLAM

Martin P. Parsley and Simon J. Julier

Abstract— **We consider how prior information can be exploited to improve the quality of SLAM. Prior information, such as aerial imagery, can be readily obtained for many environments. However, this information is often collected at a different time, using different sensors, different representations and from different vantage points than those used by the robot undertaking SLAM. In this paper, we describe a general probabilistic framework to overcome these difficulties. Our framework models the environment as a random set of latent structures which are observed by a set of sensing sytems. Each sensing system gives rise to a different kind of map and, by associating features from the same structure across the different maps, parameterised constraints between the sets of features can be constructed. These parameterised constraints make it possible to transfer information between map representations. We demonstrate the use of the framework in a simulated environment to illustrate how geometric features of different dimensions can be fused together.**

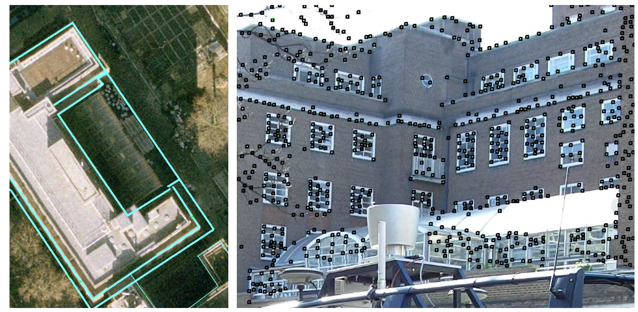

Fig. 1. Left: satellite image with line segment features extracted by hand. Right: ground view with extracted corner features, which could correspond to salient features from a 3D laser scan. These represent heterogeneous features obtained from two *different* views of the *same* building.

I. INTRODUCTION

SLAM is arguably one of the most important capabilities for an intelligent mobile platform. The ability to create and refine maps of the world in real-time is important for many tasks that range from path planning to multi-robot coordination. However, despite its apparent simplicity, SLAM poses numerous theoretial and practical problems. However, almost all SLAM research is founded on the assumption that *no* prior information is available. Although this leads to algorithms of great generality and flexibility, it does so at the cost of neglecting potentially valuable sources of information. In many situations, prior information can be readily obtained from many sources. For example, urban environments are often mapped in great detail to aid in planning or disaster management [4], [8][1]. Low-quality aerial and ground imagery is readily available through sources such as Google[2] and OpenStreetMap[3].

Some researchers have begun to develop methods to exploit this prior information. Given a "Manhattan-world" assumption, features are often assumed to cluster on planes [9], [13] and the planes can often be assumed to be locally orthogonal to one another [2]. Folkesson attempted to formalise these approaches through the development of M-space [6], which projects measurements of features into measurement subspaces. The approach has been applied to fuse visual and laser scanned data together [16]. Although these methods exploit domain knowledge about the structure of the prior information, they do not exploit the prior information itself.

The only author we are aware of who exploits the prior information directly is Kümmerle, who considered the problem of incorporating information from an aerial map into a robot using a 3D laser scanner [10]. Specifically, the aerial map was used in a Monte-Carlo Localisation-type step to improve the estimated pose of the robot which, naturally, improves the quality of the map as well. However, Monte Carlo Localisation methods treat (temporally correlated) map errors as independent sensor noises [5]. Therefore, this approach cannot handle out-of-date and incorrect maps. Furthermore, the method only handles geometric information — more general types of prior information (such as texture priors over semantic labels) might be available as well.

In this paper we consider the problem of applying prior information from a qualitatively different source to a robot performing SLAM. The structure of the paper is as follows. Challenges are outlined in Section II. In Section III we describe our parameterisation of the environment and introduce the concepts of latent structures and feature maps. Section IV presents the framework for exploiting this information. We provide a concrete implementation of this framework in Section V, showing how it can be implemented for a system performing incremental SLAM using a geometric prior map. The performance of this framework is demonstrated in simulation in Section VI, and we draw a summary and conclusions in Section VII.

This work was partially supported by a UK Research Council EPSRC Doctoral Training Award number EP/P502802/1 and partially by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence (project AA018).

M. Parsley and S. Julier are with the Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, UK M.Parsley@cs.ucl.ac.uk; S.Julier@cs.ucl.ac.uk

[1]Within the UK, for example, the Ordnance Survey MasterMap aims to map all structures within the UK to an accuracy of 1 m.

[2]http://maps.google.com; last accessed 15th July, 2010.

[3]http://www.openstreetmap.org; last accessed 15th July, 2010.

II. CHALLENGES WITH PRIOR INFORMATION

Consider the situation shown in Figure 1. A robot conducts SLAM in an urban environment using a camera or laser range finder to pick up point features on the walls of buildings. The robot has not been on the ground in the area before and thus this precludes the use of place recognition. However the same environment has also been imaged using a satellite, which collected high altitude imagery several years before. From the satellite data, the footprints of buildings can be extracted and, from these, a set of line segments determined. It is clear that this prior geometric information could be used to constrain the position of the landmarks in the map generated by the robot. However, there are several challenges in using this information:

1) The satellite data was taken from a different vantage point. As a result, it provides details about the plane of the wall, but not features on the wall.
2) The satellite data was taken using a different sensing system. Imagery data contains no information about range. Therefore, the height of the building is not known but varies across the structure. The robot, on the other hand, uses a 3D LIDAR system which supports close-range Cartesian information in 3D.
3) The signal processing and representation algorithms are different. The aerial data is processed (in this case manually) to determine a set of line segments. The robot uses point features which were extracted automatically using 3D shape descriptors.
4) The resolution of the sensors differ. In the satellite data, walls are approximated as perfectly straight lines. However, no building wall is straight. Windows are set into a wall, window ledges project from the wall. Such details can be readily detected by the robot.
5) The aerial data was taken a significant time previously. As a result, it might no longer to be an accurate representation of the state of the environment at the time the robot undertakes its operations.

To exploit the prior information, these difficulties must be overcome.

III. MODELLING PRIOR INFORMATION

A. Structures and Feature Maps

We assume that the world is populated by a set of instances of *latent structures*. The number and location of these structures are not known. Although structures could describe high-level objects in the environment (such as buildings), for our purposes they need only support part-of relationships: for example a window can be part of a wall, a wall may be part of a building, and a building may be part of a city. All structures are an instance of one of the set of *structure classes* π, $\pi \in \{1, \ldots, N\}$. For the structure class $\mathbf{s}_n$, an oracle Λ can partition the world into the set of all instances of this structure, the ith instance being denoted $\mathbf{s}_{n\{i\}}$.

Instances of structures cannot be measured directly. Instead, a sensing system (which consists of the physical sensor together with feature detection algorithms) induces certain

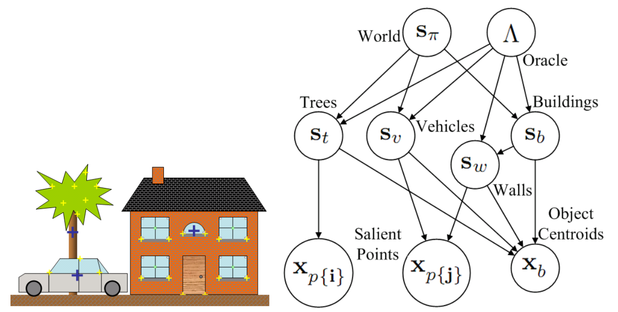

Fig. 2. Left: Example of features that may be induced from structures. The large crosses represent the centroids of the building, car and tree, while the small crosses represent salient points detected by the LIDAR. Right: Example structure class hierarchy corresponding to the top picture, defined by the oracle Λ with the feature maps $\mathbf{x}_p$ and $\mathbf{x}_b$ being produced from them, along with labels of what they represent. In this case $\mathbf{s}_v$ and $\mathbf{s}_b$ are structure classes at the same level, and $\mathbf{s}_w$ is at a level lower as it is derived from $\mathbf{s}_b$. Feature maps may be produced from instances of structure classes at all levels of the hierarchy.

features on a structure. Features must be stable, repeatedly observable, and can be represented by a state, such as a set of points in space, texture patches or the centroid of a geometric body. Different sensing systems will, in general, induce different sets of features with different dimensionalities on different parts of the same structure. In the example in Figure 1, the features in the aerial map are the lines and the features detected by the robot are points.

The features for all the sensing systems are grouped into a set of *feature maps*, $m \in \{1 \ldots M\}$ (or simply *maps*). Each map contains features of the same type and they are all parameterised the same way.

The mth feature map consists of N_m features and can be written as

$$\mathbf{x}_m = \left\{\mathbf{x}_{m\{1\}}, \mathbf{x}_{m\{2\}}, \ldots \mathbf{x}_{m\{N_m\}}\right\}$$

where $\mathbf{x}_{m\{i\}}$ is the state of the ith feature in the map.

We can generalise this to form a more complex network for several feature maps obtained from structure instances within a structure class hierarchy composed of several structure classes, as shown in Figure 2. In this example, feature maps are produced from three structure classes (buildings, trees and cars). The features within two feature maps may not be mutually exclusive, and there may be overlapping features.

A system makes observations of subsets of these features to create an estimate of a feature map.

B. Observations of Feature Maps

Consider feature map m. Given the sensing system and feature detection algorithms associated with this map, the observation model at time step k returns a set of M_{mk} observations

$$\mathbf{z}_m(k) = \left\{\mathbf{z}_{m\{1\}}, \mathbf{z}_{m\{2\}}, \ldots \mathbf{z}_{m\{M_{mk}\}}\right\}_k,$$

which are related to the map state through the observation model

$$\mathbf{z}_m(k) = \mathbf{h}_m\left(\mathbf{X}_v(k), \mathbf{x}_m, \mathbf{w}_m(k)\right),$$

where $\mathbf{X}_v(k)$ is the robot pose, and $\mathbf{w}_m(k)$ is observation noise.

Maps are inferred from a set of observations $\mathbf{z} = \mathbf{z}(1:k)$. The discrete data association parameter $\mathbf{c}(1:k)$ as commonly used in SLAM [7], which we consider part of the observation model, determines which map feature the observation goes with. Thus the estimate of the pth feature map is $p(\mathbf{x}_p|\mathbf{z}_p)$.

Having formally defined how we consider systems to produce a set of feature maps, and how there is shared information between them if they came from a common structure, we can now describe the form of this shared information, and perform inference as to whether this information applies.

IV. A FRAMEWORK FOR FUSING FEATURE MAPS TOGETHER

A. Implicit Functional Relationships Between Feature Maps

Information can be exchanged between features in different feature maps if they were generated by the same structure instance $\mathbf{s}_{w\{s\}}$. We use the binary indicator function

$$d\left(\mathbf{x}_{p\{i\}}, \mathbf{s}_{w\{s\}}\right) = \begin{cases} 1 & \text{if } \mathbf{s}_{w\{s\}} \text{ generated } \mathbf{x}_{p\{i\}} \\ 0 & \text{otherwise} \end{cases}$$

to denote if this relationship exists. To express the relationship that multiple features, such as a point i in the feature map p and a line n in feature map l were generated by the same structure, we use the expression

$$\begin{aligned} d\left(\mathbf{x}_{p\{i\}}, \mathbf{x}_{l\{n\}}, \mathbf{s}_{w\{s\}}\right) &= d\left(\mathbf{x}_{p\{i\}}, \mathbf{s}_{w\{s\}}\right) \cdot d\left(\mathbf{x}_{l\{n\}}, \mathbf{s}_{w\{s\}}\right) \\ &= d_{in,s}. \end{aligned} \tag{1}$$

This can be directly extended to any number of features and any number of feature maps. This means, for example, it can express the situation in which a single feature in one map can be associated with multiple features in another map.

If two features in two different feature maps arise from a common structure, this implies that a relationship exists between the features. We represent this relationship as a parameterised, joint implicit function [12],

$$\mathbf{f}\left(\mathbf{x}_{p\{i\}}, \mathbf{x}_{l\{n\}}\right) = \begin{cases} \theta_{in} & \text{for } d_{in,s} = 1 \\ \emptyset & \text{otherwise} \end{cases} \tag{2}$$

The parameter θ_{in} encodes any degrees of freedom that exists within the relationship. Continuing the point and line example, no wall facade is ever genuinely a flat surface. It includes, for example, intrusions due to windows and extrusions due to ledges.

The form and value of θ_{in} depends upon the choice of the joint implicit function. The prior on this parameter, $p(\theta_{in}|d_{in,s} = 1)$ serves a key role in propagating information between the maps.

This prior depends on the structure instance that generated the features, in this case $\mathbf{s}_{w\{s\}}$. However it may be the case that we do not know if $d_{in,s} = 1$; rather we may only know that $d_{in,S} = 1$, where S covers a number of structure instances including s.

An example of this is if w corresponds to building structures. Then s is a particular building which is part of a group of buildings S. We know that i and n were generated from the group of buildings, but we do not know the particular building s. In this case we wish to form a prior $p(\theta_{in}|d_{in,S} = 1)$ from the priors over the individual buildings, $p(\theta_{in}|d_{in,s} = 1)$, $s \in S$.

We can do so by marginalising over individual instances, the interpretation being that the structure instance that $\mathbf{x}_{p\{i\}}$ and $\mathbf{x}_{l\{n\}}$ were generated from is itself part of structure further up the hierarchy; this embodies the concept of "aggregation relationships" discussed in [11], and includes concepts such as wall structures being part of building structures.

Thus

$$\begin{aligned} &p(\theta_{in}|d_{in,S} = 1) = \\ &\qquad \sum_{s \in S} p(\theta_{in}|d_{in,s} = 1)\, p(d_{in,s} = 1|d_{in,S} = 1), \end{aligned} \tag{3}$$

where $p(d_{in,s} = 1|d_{in,S} = 1)$ is a prior for the features i and n being generated from a particular structure subset s, and can be conditioned on other information such as observations.

Because $p(\theta_{in}|d_{in,S} = 1)$ is in effect a weighted sum of the distributions for the individual instances we would expect it to apply to more features, but be less informative given that it is more general. Thus provided we have the distributions for the individual instances we can trade off between applying weaker constraints to more features or stronger constraints to fewer.

B. Fusion of Feature Information Between Maps

Consider the problem of fusing two features $\mathbf{x}_{p\{i\}}$ and $\mathbf{x}_{l\{n\}}$ in two feature maps p and l[4]. The maps were constructed using the observation sequences $\mathbf{z}_p$ and $\mathbf{z}_l$ respectively. To fuse these features, we need to determine the probability that the relationship holds and, given that, compute the joint distribution. Both these quantities can be derived from Bayes Rule. Suppose $d_{in,s} = 1$. In this case, the posterior probability of the joint distribution of both maps is

$$p(\mathbf{x}_m|d_{in,s} = 1, \mathbf{z}_m) = \frac{p(d_{in,s} = 1|\mathbf{x}_m)\, p(\mathbf{x}_m|\mathbf{z}_m)}{p(d_{in,s} = 1|\mathbf{z}_m)}, \tag{4}$$

where we have used the notation $\mathbf{x}_m = \{\mathbf{x}_p, \mathbf{x}_l\}$ and $\mathbf{z}_m = \{\mathbf{z}_p, \mathbf{z}_l\}$ for brevity. The probability that the features came from the same structure is the evidence in this equation, and its value is given by marginalising over $\mathbf{x}_m$,

$$p(d_{in,s} = 1|\mathbf{z}_m) = \int p(d_{in,s} = 1|\mathbf{x}_m)\, p(\mathbf{x}_m|\mathbf{z}_m) d\mathbf{x}_m.$$

Normally $d_{in,s}$ is unknown, so we would have to estimate it jointly with the state from the observations, $p(\mathbf{x}_m, d_{in,s}|\mathbf{z}_m)$, however the high dimension of the states

[4]The analysis can be extended directly to the case in which sets of features in p can be associated with sets of features in l.

makes it intractable to do so. Thus we approximate this by finding the maximum a posteriori (MAP) estimate for $d_{in,s}$,

$$d_{in,s}^{MAP} = \arg\max_{d_{in,s}} \; p(d_{in,s}|\mathbf{z}_m), \tag{5}$$

then condition the maps on this estimate using (4), estimating $p(\mathbf{x}_m|d_{in,s}^{MAP}, \mathbf{z}_m)$, in a similar manner to [3].

We can compute (5) as follows. Since we are testing the hypothesis that the sets of features are associated through the common structure instance $\mathbf{s}_{w\{s\}}$, we can rewrite this expression terms of the parameter of the joint implicit function. Specifically, note that

$$p(d_{in,s} = 1|\mathbf{x}_m) = \int p(d_{in,s} = 1|\theta_{in}) p(\theta_{in}|\mathbf{x}_m) \mathrm{d}\theta_{in}.$$

Therefore,

$$\begin{aligned} p(d_{in,s} &= 1|\mathbf{z}_m) \\ &= \int\int p(d_{in,s} = 1|\theta_{in}) p(\theta_{in}|\mathbf{x}_m) p(\mathbf{x}_m|\mathbf{z}_m) \mathrm{d}\theta_{in} \mathrm{d}\mathbf{x}_m \\ &= \int p(d_{in,s} = 1|\theta_{in}) p(\theta_{in}|\mathbf{z}_m) \mathrm{d}\theta_{in} \end{aligned}$$

We can transform this expression to be in terms of the prior of the parameter through the application of Bayes Rule again. Specifically,

$$p(d_{in,s} = 1|\theta_{in}) = \frac{p(\theta_{in}|d_{in,s} = 1) p(d_{in,s} = 1)}{p(\theta_{in})},$$

where

$$\begin{aligned} p(\theta_{in}) = {} & p(\theta_{in}|d_{in,s} = 1) p(d_{in,s} = 1) \\ & + p(\theta_{in}|d_{in,s} = 0) p(d_{in,s} = 0). \end{aligned}$$

The distribution $p(\theta_{in}|d_{in,s} = 1)$ is the prior determined by the relationship with the structure. Since θ_{in} is not defined when $d_{in,s} = 0$, we use the uniformative likelihood $p(\theta_{in}|d_{in,s} = 0) = 1$.

Thus, if we find that $p(d_{in,s} = 1|\mathbf{z}_m)$ is significantly probable (where this depends on the application), we can decide that $d_{in,s} = 1$ and condition the state on it.

V. INCREMENTAL SLAM IMPLEMENTATION

In this section we describe how the framework can be implemented in a full covariance, EKF-based SLAM algorithm.

Consider the motivating example again from Figure 1. Suppose $\mathbf{x}_{p\{i\}}$ is a point which might lie on the wall of a building, and $\mathbf{x}_{l\{n\}}$ is a model of the walls of the building. $\mathbf{x}_{l\{n\}}$ is parameterised by its two end points $\mathbf{x}^1_{l\{n\}}$ and $\mathbf{x}^2_{l\{n\}}$. In this case, θ_{in} is the orthogonal distance of $\mathbf{x}_{p\{i\}}$ from $\mathbf{x}_{l\{n\}}$. Its value is computed from

$$\theta_{in} = \frac{\det\left(\left[\mathbf{x}^2_{l\{n\}} - \mathbf{x}^1_{l\{n\}} \quad \mathbf{x}^1_{l\{n\}} - \mathbf{x}_{p\{i\}}\right]\right)}{\left|\mathbf{x}^2_{l\{n\}} - \mathbf{x}^1_{l\{n\}}\right|}.$$

If $\mathbf{x}_{p\{i\}}$ and $\mathbf{x}_{l\{n\}}$ were generated by the same structure and the model for $\mathbf{x}_{l\{n\}}$ were correct, then $\theta_{in} = 0$. However, as explained earlier, these models are not correct and thus this will generally not be the case. Therefore, we allow for some uncertainty.

We have discussed detailed implementations elsewhere [14]; here we focus on the calculation of the joint distribution and inferring the structure.

A. Inferring the Structure

We first need to determine the probability $p(d_{in,s} = 1|\mathbf{z}_m)$. Although it is possible to use multiple hypothesis tracking to account for the evolution of this quantity over time [15], we have found that very similar performance can be achieved at a lower cost by delaying the application of constraints until $p(d_{in,s} = 1|\mathbf{z}_m) > p_{accept}$, an acceptance threshold.

We compute this as follows. At time step k, Mahalanobis gating is first applied to identify the subset of J features $I = \{i_1, \ldots, i_J\}$ in feature map p, $\mathbf{x}_{p\{I\}}$ that potentially relate to a set of map line segment features $N = \{n_1, \ldots, n_L\}$. The mapping is not one-to-one: several points can be associated with the same line.

Thus we have a vector of indicators which evaluate to a binary vector D, where the jth element is

$$D^j = d\left(\mathbf{x}_{p\{I^j\}}, \mathbf{x}_{l\{N^j\}}, \mathbf{s}_{w\{S^j\}}\right).$$

Assuming Gaussianity, the likelihood corresponding to the set of elements M, D^M is given by

$$p(\theta^M_{IN} = 0|\mathbf{z}) = \left((2\pi)^{\frac{c}{2}}\sqrt{|\mathbf{S}|}\right)^{-1} \exp\left(-\frac{1}{2}\nu^T\mathbf{S}^{-1}\,\nu\right)$$

where c is the dimension of ν, ν and $\mathbf{S}$ are the innovation and innovation covariance, $\nu = \mathbf{f}(\mathbf{x}_{p\{I^M\}}, \mathbf{x}_{l\{N^M\}})$ and $\mathbf{S} = \nabla\mathbf{f}\,\mathbf{P}_m\,\nabla^T\mathbf{f}$, where $\nabla\mathbf{f}$ is the Jacobian of $\mathbf{f}(\cdot)$.

We seek the most probable estimate for D, D_{MAP} by permuting what happens within l and s. We enumerate over all combinations of D to obtain a set of hypotheses $D_1 \ldots D_N$, and then compute the probability distribution $p(D_J|\mathbf{z}_p(1 : t), \mathbf{z}_l)$ over these hypotheses. In our EKF-SLAM based implementation, we cannot revise incorrectly applied constraints. Therefore, it is important to ensure that D_{MAP} is sufficiently probable that it is likely to be the correct hypothesis. We have found that $p_{accept} = 0.7$ is sufficient for this case.

In our implementation we use a constant prior for $p(d\left(\mathbf{x}_{l\{n\}}, \mathbf{s}_{w\{s\}}\right) = 1)$, the probability that wall features come from walls (encompassing spurious features), where the s wall structure instance is defined in terms of producing line $\mathbf{x}_{l\{n\}}$. We also use a constant for $p(d(\mathbf{x}_{p\{i\}}, \mathbf{s}_{w\{s\}}) = 1|d(\mathbf{x}_{l\{n\}}, \mathbf{s}_{w\{s\}}) = 1)$, the probability that the point $\mathbf{x}_{p\{i\}}$ came from the sth structure instance defined as having generated the line segment $\mathbf{x}_{l\{n\}}$ (with $p(d(\mathbf{x}_{p\{i\}}, \mathbf{s}_{w\{s\}}) = 1|d(\mathbf{x}_{l\{n\}}, \mathbf{s}_{w\{s\}}) = 0) = 0$). Furthermore we assume that the structure instance heritage of independent features is independent.

$P(d_{in,s} = D)$ can then be inferred from the structure of the hierarchy as products of these density functions, considering that

$$P(d_{in,s} = D) = p(d_{I,S} = A | d_{N,S} = B)\, p(d_{N,S} = B),$$

where $A\,B = D$.

For instance for the set of K points $\mathbf{x}_{p\{K\}}$ that may come from the same wall structure $\mathbf{s}_{w\{s\}}$ as the nth line segment $\mathbf{x}_{l\{n\}}$,

$$P(d_{Kn,s} = D) = \begin{cases} p(d_{n,s}{=}0){+}p(d_{n,s}{=}1) \prod_{j \in K} p(d_{j,s}{=}0|d_{n,s}{=}1) & \text{if } D{=}\mathbf{0} \\ p(d_{n,s}{=}1) \prod_{j \in K} p(d_{j,s}{=}D^j|d_{n,s}{=}1) & \text{otherwise} \end{cases}.$$

Although we have used constant priors, we could condition these on observations through the use of classifiers.

Having computed the probabilites of all hypotheses $D_1 \ldots D_N$, we accept D_{MAP} (the mode) if its probability exceeds an upper threshold p_{accept}.

We can attempt to form such a hypothesis by marginalising over the hypotheses of some features (in effect they are removed from all D), and thus we do not attempt to determine whether a common structure holds in their case. If the most probable hypothesis is that $D = \mathbf{0}$ then we do not apply any constraint information, and repeat the process at a future time step.

B. Computing the Joint Distribution

Given that $d_{in,s} = 1$, we evaluate (4) as follows. At timestep k $p(\mathbf{x}_p|\mathbf{z}_p(1:k))$ and $p(\mathbf{x}_l|\mathbf{z}_l)$ are the map estimates from the SLAM state and prior map respectively; here the observations $\mathbf{z}_p(1:k)$ and $\mathbf{z}_l$ are assumed independent however this condition is not required. In the EKF these are represented by their mean $\hat{\mathbf{x}}_p, \hat{\mathbf{x}}_l$ and covariance $\mathbf{P}_p, \mathbf{P}_l$. The a priori joint map estimate is

$$\hat{\mathbf{x}}_m = \begin{bmatrix} \hat{\mathbf{x}}_p \\ \hat{\mathbf{x}}_l \end{bmatrix}, \mathbf{P}_m = \begin{bmatrix} \mathbf{P}_p & 0 \\ 0 & \mathbf{P}_l \end{bmatrix}.$$

Then (4) constitutes a pseudo-observation [14] between some features in $\mathbf{x}_m$. The innovation $\nu = 0 - \mathbf{f}(\hat{\mathbf{x}}_{p\{i\}}, \hat{\mathbf{x}}_{l\{n\}})$, with covariance $\mathbf{S} = \nabla\mathbf{f}\,\mathbf{P}_m\,\nabla^T\mathbf{f}$, where $\nabla\mathbf{f}$ is the Jacobian of $\mathbf{f}(\cdot)$. A Kalman update then gives the posterior joint map estimates $p(\mathbf{x}_p, \mathbf{x}_l | d_{in,s} = 1, \mathbf{z}_p(1:k), \mathbf{z}_l)$,

$$\mathbf{K} = \mathbf{P}_m\,\nabla^T\mathbf{f}\,\mathbf{S}^{-1},$$
$$\hat{\mathbf{x}}_m^+ = \hat{\mathbf{x}}_m + \mathbf{K}\,\nu,$$
$$\mathbf{P}_m^+ = \mathbf{P}_m - \mathbf{K}\,\mathbf{S}\,\mathbf{K}^T,$$

where $\hat{\mathbf{x}}^+$ and $\mathbf{P}^+$ are the posterior mean and covariance. The joint map estimate can be represented in a way that is more robust to linearisation and other errors [14].

VI. Simulation Studies

We have used simulations to investigate the effect of using a geometric prior map to aid a vehicle performing EKF-SLAM in a large urban environment, as used by [14] (we also use their Dual Representation form). We chose the EKF because it is well studied and maintains an explicit covariance matrix. Although several alternatives to the EKF could be used, they suffer from their own drawbacks, such as inconsistency in the case of FastSLAM [1].

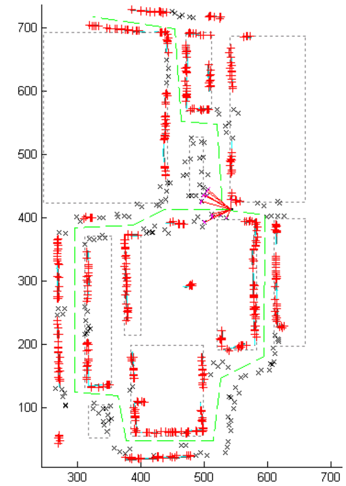

Fig. 3. The simulated environment, units in m. The dashed line shows the trajectory, which starts near the centre and completes an anti-clockwise loop before ending at the top. The dashed rectangles show occlusion boundaries.

The robot is a steered bicycle (with velocity and steer 1σ errors of 0.5 m and 3°) equipped with a range-bearing sensor (with 0.01 m and 1.15° 1σ errors), starting at a known location and travelling at 14 m/s along the 1.4 km trajectory shown in Figure 3. The SLAM system builds up a 2D metric map of point beacons, and has a noisy prior map with a 1 m std. dev. error consisting of line segment feature estimates representing the outer surfaces of building walls $\mathbf{s}_w$, as can be obtained from OS MasterMap. The structure between the two maps results in points lying on line segments, in which case the normal distance is 0; however 40% of the beacons in the environment lie close to but not on the building walls, and 11% of the line segments in the prior map are spurious; thus the algorithm must determine which features have a common structure before it can use this to inform the SLAM process (otherwise the estimate will be erroneous). We use $p(d\left(\mathbf{x}_{l\{n\}}, \mathbf{s}_{w\{s\}}\right) = 1) = 0.89$ and $p(d\left(\mathbf{x}_{p\{i\}}, \mathbf{s}_{w\{s\}}\right) = 1 | d\left(\mathbf{x}_{l\{n\}}, \mathbf{s}_{w\{s\}}\right) = 1) = 0.6$. We use $p_{accept} = 0.7$.[5]

The results are presented for the average over 30 Monte Carlo runs. Because we do not use map-management techniques such as submapping, for computational reasons we ran the filter as a sliding window, removing beacons not seen for 6 seconds (84 m robot travel distance).

Figure 4 shows the absolute error in the robot pose estimate, along with the 3σ bound of the error estimate. The SLAM system using the prior map shows a dramatic improvement in consistency and accuracy from the prior information, showing that relationships can be determined robustly by computing the posterior probability of their holding, and thus a geometric prior map can be used to

[5] This optimistic threshold gave the best results in our case; we expect that this is because of the inconsistency exhibited by regular EKF-SLAM.

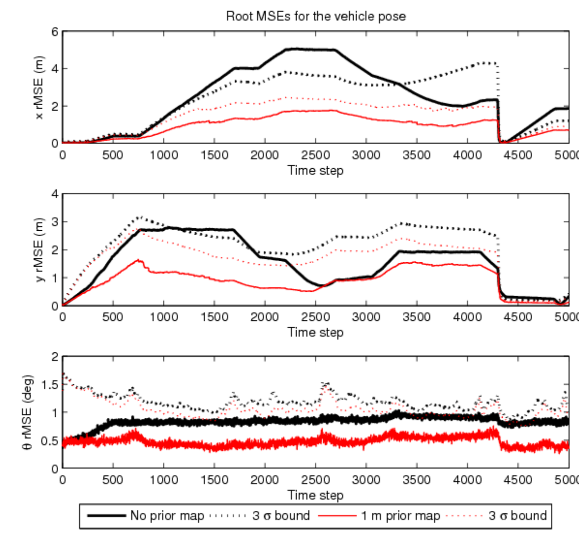

Fig. 4. Robot x, y and θ pose error over the trajectory for regular SLAM without a prior map, and with a prior map of 1 m std. dev. error.

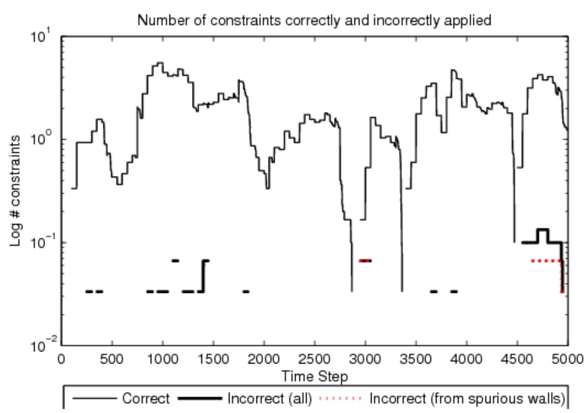

Fig. 5. Log number of constraints applied correctly (thin line) and incorrectly (thick line) to features in the state, with the thin dotted line indicating how many were from spurious line segments. Because we use a sliding window features are constantly being added and removed from the state. Missing values indicate that no incorrect constraints were applied.

significantly improve SLAM, even when the 1σ error in the prior map is 1 m (compared with the 0.01 m range accuracy of the sensor).

Figure 5 shows the number of constraints applied correctly (there is common structure and thus the constraint holds) and incorrectly. For most of the trajectory no constraints were applied incorrectly (as indicated by the missing values), however for the few that were, the number applied correctly was between one and two orders of magnitude higher on average, showing that the method is effectively able to determine when there is common structure.

VII. SUMMARY AND CONCLUSIONS

In this paper we have presented a framework for exploiting prior information in SLAM. The key challenge arises when the prior information is of a different form. To overcome these difficulties, we introduced the notion of latent structures and the fact that multiple heterogeneous types of information can be regarded as different feature maps.

We applied the framework to show that a robot performing point-based metric SLAM can use a prior map of line segments to significantly improve its estimate, by inferring the features for which an underlying structure holds, then using this to exploit the information from the prior map.

We are currently developing an implementation of the framework and applying it to a 3D data set. In the future, we plan to greatly extend the types of features which are supported to include semantic labels, and to expand the range of constraints to include topological constraints as well.

REFERENCES

[1] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *Robotics and Automation, 2006. ICRA 2006. Proc. 2006 IEEE International Conference on*, 2006, pp. 424–429.

[2] K. R. Beevers and W. H. Huang, "Inferring and Enforcing Relative Constraints in SLAM," in *2006 Workshop on the Algorithmic Foundations of Robotics (WAFR 2006)*, New York, NY, July 2006.

[3] C. Bibby and I. Reid, "Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with Reversible Data Association," in *Proceedings of Robotics Science and Systems*, 2007.

[4] C. Brenner, "Towards Fully Automatic Generation of City Models," in *In: IAPRS*, 2000, pp. 85–92.

[5] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo Localization for Mobile Robots," in *IEEE International Conference on Robotics and Automation (ICRA99)*, May 1999.

[6] J. Folkesson, P. Jensfelt, and H. Christensen, "The M-space Feature Representation for SLAM," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1024–1035, October 2007.

[7] U. Frese, "A Discussion of Simultaneous Localization and Mapping," *Auton. Robots*, vol. 20, no. 1, pp. 25–42, 2006.

[8] C. Fruh and A. Zakhor, "An Automated Method for Large-Scale, Ground-Based City Model Acquisition," *International Journal of Computer Vision*, vol. 60, pp. 5–24, 2004.

[9] A. P. Gee, D. Chekhlov, W. Mayol, and A. Calway, "Discovering Planes and Collapsing the State Space in Visual SLAM," in *Proc. the 18th British Machine Vision Conference*, September 2007.

[10] R. Kummerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, "Large scale graph-based SLAM using aerial images as prior information," in *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[11] B. Limketkai, L. Liao, and D. Fox, "Relational Object Maps for Mobile Robots," in *IJCAI*, 2005, pp. 1471–1476.

[12] J. Neira and J. Tardos, "Data Association in Stochastic Mapping Using the Joint Compatibility Test," *Robotics and Automation, IEEE Trans.*, vol. 17, no. 6, pp. 890–897, 2001.

[13] A. Nuchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann, "3d mapping with semantic knowledge," in *In RoboCup International Symposium*, 2005, pp. 335–346.

[14] M. Parsley and S. Julier, "SLAM with a Heterogeneous Prior Map," in *Proceedings of the Fourth Annual SEAS DTC Conference*, 2009.

[15] M. P. Parsley and S. J. Julier, "The Common State Filter for SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2060–2065.

[16] A. J. B. Trevor, J. G. Rogers, C. Nieto, and H. I. Christensen, "Applying Domain Knowledge to SLAM using Virtual Measurements," in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA2010)*, Anchorage, AK, USA, 3–8 May 2010, pp. 5389–5394.