2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

Exploiting Prior Information in GraphSLAM

Martin P. Parsley and Simon J. Julier

***Abstract*— In this paper we present a general method for exploiting prior information to constrain the location of landmarks in GraphSLAM. Prior information can be obtained for many environments in many different ways. However, this information can be incomplete, out-of-date, or presented in a different form than that used by the robot. Therefore, we argue that prior information is most naturally modelled as sets of potential constraints that act between landmarks. We present an extension of GraphSLAM that incorporates these constraints. We illustrate the results in an experiment with a 3D laser scanner and demonstrate a significant improvement in performance.**

I. Introduction

Simultaneous localisation and mapping (SLAM) is one of the most important and established techniques in mobile robotics. The ability to simultaneously navigate and map an environment is important for many tasks. Almost all SLAM research is founded on the assumption that *no* prior information is available. Although this leads to general algorithms of great flexibility and capability, it does so at the cost of ignoring readily available information. For example, low-quality aerial and ground imagery is readily available through online sources such as Google[1] and OpenStreetMap[2] for much of the globe. Furthermore many cities are mapped in great detail to aid in planning or disaster management [1], [2][3]. This prior information contains geometric and semantic information, and could be used to greatly improve the performance of SLAM algorithms.

Several authors have begun to explore how prior information can be used. The use of *local* information, such as the planarity of features on planes [3] and orthogonality of walls [4], has been shown to significantly improve the quality of the computed solutions. More recently, authors have begun to consider how *absolute* information can be exploited as well [5]–[9]. At first sight this is a simple process: where global information exists, the SLAM problem can be transformed into a localisation problem using a known map, and techniques such as Kalman filters or Monte Carlo localisation can be used. However, there are several difficulties with this approach. First, the prior information (an aerial map) might be created using sensing systems and vantage points very different from those used by the robot for purposes other than localisation. As a result, the types and kinds of features stored might be different from the internal representations used by the robot. Second, the prior information could be collected at a different time and might no longer be up-to-date [5]. As a result, prior absolute information requires both data association (to identify where the prior information can be applied), and transformations between map representations, to apply complete or partial information.

平面度

In this paper, we consider the problem of how prior information can be integrated into the GraphSLAM algorithm. We argue that because of limitations and errors, prior information is *not* equivalent to a known map. Rather, it should be treated as a set of constraints between feature vectors. By doing so, the effects of prior information can be modelled and tracked on a feature-by-feature basis.

The structure of the paper is as follows. The problem statement is introduced in Section II. In section III, we present our extension to GraphSLAM to probabilistically introduce the effects of prior information. Unlike previous approaches, we argue that the effects of prior information should be introduced at the feature-level. The implementation of this algorithm for a vehicle performing 3D SLAM with a vehicle-mounted LIDAR scanner is presented in Section IV. Results are discussed in Section V and a summary and conclusions presented in Section VI.

II. SLAM

A. Formulation of the SLAM Problem

The goal of a SLAM algorithm is to estimate the pose of a robot at time step k, $\mathbf{x}_v(k)$, together with the map estimate $\mathbf{x}_p(k)$, the map with form p consisting of a number of observable *features* such as points in 3D space, some of which are observed from multiple poses. The kth pose is related to the $k-1$th by the state transition model

$$\mathbf{x}_v(k) = F(\mathbf{x}_v(k-1), \mathbf{u}(k), \mathbf{v}(k)),$$

where $\mathbf{u}(k)$ is odometry and $\mathbf{v}(k)$ is zero-mean noise with known covariance $\mathbf{Q}$. This transition function determines how the platform moves from one pose to the next (the features in the map are assumed static).

At time k, a set of observations $\mathbf{z}(k)$ are made, each of which is related to the map by the observation function

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{c}(k), \mathbf{w}(k)),$$

where $\mathbf{c}(k)$ is a discrete association parameter that indicates which features are being observed, and $\mathbf{w}(k)$ is zero-mean noise with known covariance $\mathbf{R}$.

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

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

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

978-1-61284-385-8/11/$26.00 ©2011 IEEE

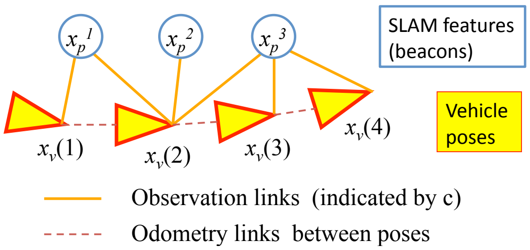


Fig. 1. Graphical model of GraphSLAM showing the poses, map features and links between them.

B. GraphSLAM

GraphSLAM transforms SLAM into the problem of computing a maximum likelihood estimate over a graph [10]. The structure of the graph is shown in Fig. 1: it is composed 信标 of vehicle pose nodes, beacon position nodes, and the links between them. Poses link to one another via odometry. Beacons are linked to poses through data association.

More formally, given the time history of platform poses $\mathbf{x}_v = \mathbf{x}_v(1:k)$, control inputs $\mathbf{u} = \mathbf{u}(1:k)$, observations $\mathbf{z} = \mathbf{z}(1:k)$ and association variables $\mathbf{c} = \mathbf{c}(1:k)$, GraphSLAM computes the maximum a posteriori state

$$\hat{\mathbf{x}}_{ML} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{c},\mathbf{z},\mathbf{u}).$$

Given an uninformative prior and the conventional assumption that all the marginal distributions are Gaussian, it can be shown that this is equivalent to solving the nonlinear least squares estimation problem [10],

$$\hat{\mathbf{x}}_{ML} = \arg\min_{\mathbf{x}} \Bigg(\|\mathbf{x}_v(1) - \hat{\mathbf{x}}_v(1)\|^2_{\mathbf{P}_1} + \|\mathbf{h}(\mathbf{x},\mathbf{c}) - \mathbf{z}\|^2_{\mathbf{R}} + \sum_{k=2}^{K} \|F(\mathbf{x}(k-1),\mathbf{u}(k)) - \mathbf{x}(k)\|^2_{\mathbf{Q}} \Bigg), \qquad (1)$$

where $\|v\|^2_{\mathbf{P}} = v^T \mathbf{P}^{-1} v$ is a weighted mean squared error.

Each term models a different constraint: the first accounts for initial conditions, the second for the observation sequence (via the observation model), and the third for the control inputs (via the process model).

C. Introducing Prior Information into GraphSLAM

One of the most attractive features of GraphSLAM is that it offers a very explicit way in which extra constraints — in the form of extra cost terms — can be incorporated into the mapping equation 1. As a result, a number of authors have considered how prior information can be incorporated. There are two basic approaches — those based on constraining the location of the platform, and those based on constraining the locations of the features.

In the first category, Kümmerle [9] and Karg [11] consider how absolute prior information (a satellite plan of a city and a floor plan of a building respectively) can be used to constrain the location of the platform. Their approach is illustrated in Fig. 2: assuming the prior map (denoted $\mathbf{x}_m$) is perfect, Monte Carlo localisation is used to compute

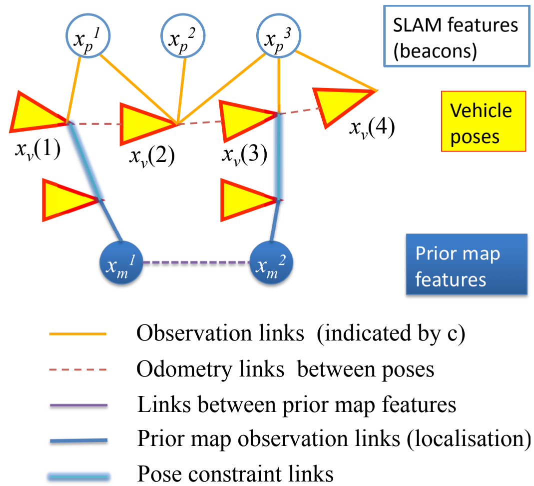


Fig. 2. Graphical model showing the integration of a prior map indirectly into GraphSLAM via intermediate platform poses, similarly to [9].

a set of localised vehicle states. These vehicle states are used to constrain the vehicle poses in the SLAM algorithm. However, the errors in the localised vehicle states are *not* independent for a number of reasons. The most significant is that the method does not account for the fact that the prior map is imperfect. The prior map was constructed using another mapping process and is subject to its own sources of errors. As a result, the map can be incomplete (miss features), contain spurious features, or contain features whose position errors can be correlated over a map. These error sources manifest themselves as the thick line in Fig. 2 between the prior map features. Current approaches, however, neglect these dependencies presumably on the assumption that the map errors are small. The second source of dependence is that the *same* observation information is used in both mapping and prior map exploitation.

In the second category, the only work we are aware of is that by Trevor, who proposed the use of "virtual measurements" to incorporate domain knowledge (such as the effects of walls) [12]. However a prior map is not explicitly used, and the problem of associating features between the SLAM state and prior map is not addressed.

Takemura et al. [13] discuss how a universal map could be used to aid localisation and navigation. However the map is not considered as part of the SLAM process, and it is not clear how such a universal map would be created and represented in a way that would make it able to represent the large variety of possible maps, each with their own correlated error structures.

III. Prior Information as Constraints on Features

We take a different approach, where instead of linking features from the prior map through vehicle poses, we directly link such features to features in the state.

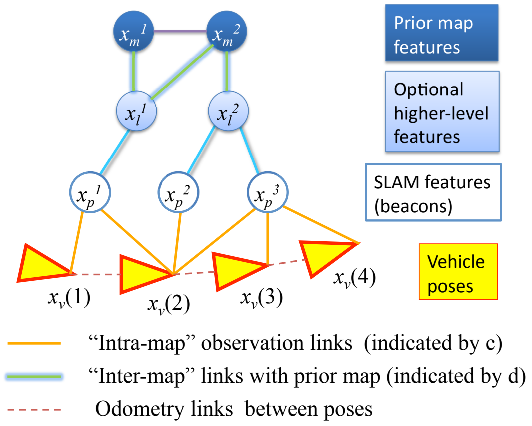


Fig. 3. Graphical model showing the approach in this paper for integrating a prior map into GraphSLAM directly through feature associations, showing constraints between the (optional) higher-level SLAM-derived features and those of the prior map.

A. Applying Constraints to Features

With reference to Fig. 1, we link features in the prior map to features in the SLAM map through inter-map constraints, as shown in Fig. 3. The associations for these 什么是observation? links are denoted by a discrete parameter $\mathbf{d}$, in the same way that observations are linked to features through the parameter $\mathbf{c}$. Considering $\mathbf{z} = \{\mathbf{z}_m, \mathbf{z}_p\}$, the joint state is $p(\mathbf{x}_v, \mathbf{x}_p, \mathbf{x}_m | \mathbf{d}, \mathbf{c}, \mathbf{z}, \mathbf{u})$, where the p and m subscripts denote the GraphSLAM and prior map respectively, the prior map estimate being $p(\mathbf{x}_m | \mathbf{z}_m)$.

The prior map can be exploited in GraphSLAM if there is a parameterised relationship between any features in the prior map and state [5]. The existence of such a relationship implies that

$$\mathbf{f}(\mathbf{x}_p, \mathbf{x}_m, \mathbf{d}) = \boldsymbol{\theta}, \tag{2}$$

where $\mathbf{x}_p$ is the SLAM map, $\mathbf{x}_m$ is the prior map, $\mathbf{f}(\cdot)$ is the parameterisation of the relationship between the features in the state and prior map indicated by $\mathbf{d}$, and $\boldsymbol{\theta}$ is a parameter. These constraints constitute the links between the SLAM features and prior map features in Fig. 3. Because it is not clear whether a constraint applies between a feature in the state and prior map, both the form of $\mathbf{f}(\cdot)$, and the value of $\boldsymbol{\theta}$ are latent. However, conditioned on a form for $\mathbf{f}(\cdot)$, a prior for $\boldsymbol{\theta}$ can be established from training data.

Assuming Gaussianity, the maximum likelihood estimate of both maps can be estimated by introducing two additional constraints to (1), $\left\| \mathbf{f}(\mathbf{x}_p, \mathbf{x}_m, \mathbf{d}) - \hat{\boldsymbol{\theta}} \right\|^2_{\mathbf{P}_\theta}$ and $\|\mathbf{x}_m - \hat{\mathbf{x}}_m\|^2_{\mathbf{P}_m}$, where the first term is a constraint based on the parameterisation of the relationship between the state and prior map, and the second term accounts for the initial prior map estimate $\hat{\mathbf{x}}_m$, with covariance $\mathbf{P}_m$.

B. Higher-level features

Many prior maps represent high level structures in the environment, such as polygons representing building walls. However in most cases the SLAM state represents the environment by many low level features, such as points from laser scans. Reasoning with such low level features can be computationally inefficient, due to the high dimension of the state and the high number of features. In such cases a smaller state of higher-level features can be extracted from the SLAM state, a strategy we take in our implementation.

Such higher-level features correspond to the same kind of underlying structures as those in the prior map. For instance a dense map of scan points can be converted into one of planes if the prior map predominantly represents planar structures [14]. Thus scan points that were clearly generated from non-flat structures would not have to be considered further. Ultimately higher level structures such as buildings can be detected and represented [5], [15], [16], allowing for exploitation of a greater variety of prior maps, such as tourist maps.

Higher level features constitute intermediate nodes linking the low-level features to features in the prior map, as shown in Fig. 3, and are extracted from lower level features in the state as follows. Given a map from GraphSLAM, $p(\mathbf{x}_p | \mathbf{c}, \mathbf{z}, \mathbf{u})$, where $\mathbf{x}_p$ is the GraphSLAM map of low-level features, a map of higher-level features $\mathbf{x}_l$ is obtained via a model $p(\mathbf{x}_l | \mathbf{x}_p)$. Then the map of higher level features is

$$p(\mathbf{x}_l | \mathbf{c}, \mathbf{z}, \mathbf{u}) = \int p(\mathbf{x}_l | \mathbf{x}_p)\, p(\mathbf{x}_p | \mathbf{c}, \mathbf{z}, \mathbf{u})\, \mathrm{d}\mathbf{x}_p. \tag{3}$$

For computational efficiency this can be implemented by extracting $\hat{\mathbf{x}}_l$ from the maximum likelihood map $\hat{\mathbf{x}}_p$ in (5), in which case $p(\mathbf{x}_l | \hat{\mathbf{x}}_p)$ is a delta function, and is the method we use in our implementation. More advanced methods of detecting and extracting features associated with underlying structures (such as cars) from point clouds could be used instead [16].

C. Measurement and Prior Information Association

Usually the data association parameter $\mathbf{c}$ is unknown and is chosen to maximise $p(\mathbf{x}, \mathbf{c} | \mathbf{z})$. However this form is computationally intractable because of the high dimension of $\mathbf{c}$ and $\mathbf{x}$. Therefore we use a computationally efficient approximation to EM to iteratively estimate $\mathbf{c}$ and $\mathbf{x}$, in a similar manner to [17], [18]. The algorithm iterates between the two steps

$$\hat{\mathbf{c}}_{ML} = \arg\max_{\mathbf{c}} p(\mathbf{c} | \mathbf{x}), \tag{4}$$

$$\hat{\mathbf{x}}_{ML} = \arg\max_{\mathbf{x}} p(\mathbf{x} | \mathbf{c}, \mathbf{z}, \mathbf{u}). \tag{5}$$

The initial map is obtained from odometry and projecting the observations into features. The inter-map association parameter $\mathbf{d}$ can similarly be estimated from the maps by iterating

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d}} p(\mathbf{d} | \mathbf{c}, \mathbf{x}) \tag{6}$$

$$\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_l, \hat{\mathbf{x}}_m = \arg\max_{\mathbf{x}_v, \mathbf{x}_l, \mathbf{x}_m} p(\mathbf{x}_v, \mathbf{x}_l, \mathbf{x}_m | \mathbf{d}, \mathbf{c}, \mathbf{z}, \mathbf{u}). \tag{7}$$

Computationally there are a growing number of techniques which could be used to solve these efficiently; (5) and (7)

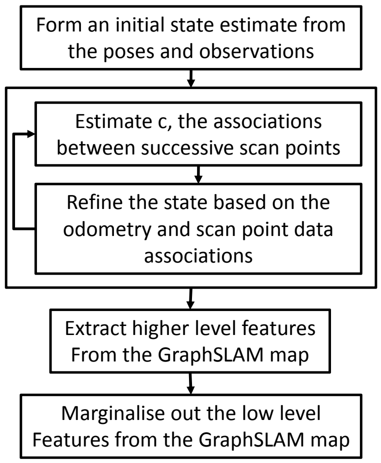


Fig. 4. Stages in the production of a map of sparse higher level features using GraphSLAM

can be solved using [19], whereas (4) and (6) can be solved approximately using methods such as scan-matching [20] and Joint Compatibility Branch and Bound (JCBB) [21].

IV. Implementation

In our implementation a vehicle performs SLAM with a 3D laser scanner, producing a map of points $\mathbf{x}_p$. The prior map $\mathbf{x}_m$ consists of planes representing flat vertical building walls, and thus we extract a map of planes $\mathbf{x}_l$ from the GraphSLAM map of points, then associate these with the planes in the prior map. We perform the estimation in several steps. We first produce a joint state consisting of the vehicle poses and sparse high level features from GraphSLAM by iterating (4)-(5) then applying (3), then fuse this map with the prior map by iterating (6)-(7) to obtain improved estimates of the trajectory and maps.

The stages involved in the production of a map of sparse high level features using GraphSLAM are shown in Fig. 4, and those involved in the fusion of the GraphSLAM map with the prior map are shown in Fig. 5.

A. GraphSLAM stage

Given the observation association parameter $\mathbf{c}$ and/or the inter-map association parameter $\mathbf{d}$, we compute (5) and (7) from (1), using well-known nonlinear least squares estimation [17], [22].

This involves iteratively forming the Jacobians associated with the transition and observations function $f(\cdot)$ and $\mathbf{h}(\cdot)$ in (1), and concatenating them into a matrix $\mathbf{J}$, the information matrix and innovation associated with these constraints being $\mathbf{Y}$ and $\mathbf{v}$ respectively, as shown in [17]. Then we can solve for δ in

$$\mathbf{J}^T \mathbf{Y} \, \mathbf{J} \, \delta = \mathbf{J}^T \, \mathbf{Y} \, \mathbf{v}, \tag{8}$$

δ being an increment added to the estimate on the ith iteration, $\hat{\mathbf{x}}_{ML}^{i+1} = \hat{\mathbf{x}}_{ML}^{i} + \delta$. We solve this using direct sparse methods [23]. The corresponding information matrix of the new estimate $\hat{\mathbf{x}}_{ML}$ is given by $\mathbf{Y}_{ML} = \mathbf{J}^T \mathbf{Y} \, \mathbf{J}$ [17].

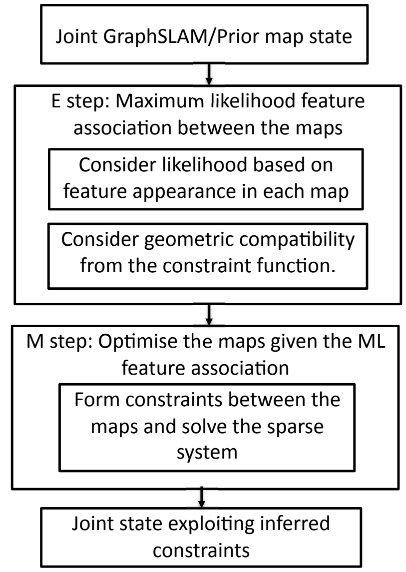


Fig. 5. Stages in the fusion of the GraphSLAM map with the prior map in our implementation.

In our implementation we estimate the parameter for the associations between successive scans, $\mathbf{c}$ in (4) using Euclidean nearest-neighbour association, however scan-matching [20] could be used instead.

B. Extraction of higher level features from the GraphSLAM map

We implement (3), extracting planes from the maximum likelihood map estimate (which consists of 3D points) using RANSAC. Because our prior map consists of walls, this can be considered a heuristic-based implementation of wall structure detection and extraction. Thus point features that obviously do not correspond to walls (for instance they were produced by vegetation) can be discarded.

We integrate identified planes into the state by minimising the orthogonal distance between each plane and its constituent points using (8). Each plane is a rectangular surface parameterised by three vertices, their position in the plane being determined by fitting a minimum-area rectangle to encompass its constituent points (when projected onto the plane).

Having integrated the planes into the state, we marginalise out the points using the method in [10], leaving a state consisting of a set of poses and planes, which we can associate with planes from the prior map.

C. Prior map integration stage

The prior map state (mean $\hat{\mathbf{x}}_m$ and covariance $\mathbf{P}_m$) is appended to the GraphSLAM state to form a joint state consisting of all the planes and poses. We then iterate (6) and (7). We estimate (6), the maximum likelihood association parameter $\mathbf{d}$ between planes in the two maps as follows.

For computational efficiency, we use gating to obtain an initial association matrix to reject plane segments which are obviously unlikely to associate together. Thus planes that

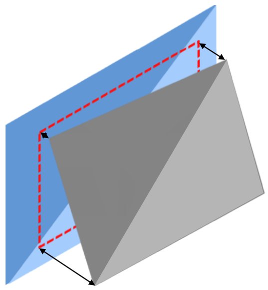


Fig. 6. Orthogonal distance between the three vertices making up one plane and the other plane. If the two planes associate, the error of each such distance is distributed with mean $\hat{\theta}$ and variance P_θ.

are too far from each other and have a high dihedral angle, compared to the magnitude of their errors are not considered to gate.

We then refine the association matrix probabilistically. We have found that for our data set JCBB [21], where the likelihood function is based on the Mahalanobis distance and number of associations is sufficient, however other likelihood models could be used. The constraint form $\mathbf{f}(\cdot)$ from (2) that we use is that the normal distance of the three vertices of one of the planes to the corresponding plane, shown by the arrows in Fig. 6, is distributed with known mean $\hat{\theta} = \mathbf{0}$ and covariance $\mathbf{P}_\theta = \text{diag}(\begin{bmatrix} 0.25 & 0.25 & 0.25 \end{bmatrix})$ (we have estimated these, however they would normally be derived from analysis of training data, such as a set of accurate maps),

$$\mathbf{f}(\mathbf{x}_l, \mathbf{x}_m, \mathbf{d}) = \begin{bmatrix} \hat{\mathbf{n}}.\mathbf{x}_l^1 & \hat{\mathbf{n}}.\mathbf{x}_l^2 & \hat{\mathbf{n}}.\mathbf{x}_l^3 \end{bmatrix}^T + q,$$

where $\mathbf{x}_l^n$ is the nth vertex of a plane in the map l that is associated with a plane in $\mathbf{x}_m$ whose Hessian normal form is $\hat{\mathbf{n}}.\mathbf{x}_m = -q$, where

$$\hat{\mathbf{n}} = \frac{(\mathbf{x}_m^2 - \mathbf{x}_m^1) \times (\mathbf{x}_m^3 - \mathbf{x}_m^1)}{|(\mathbf{x}_m^2 - \mathbf{x}_m^1) \times (\mathbf{x}_m^3 - \mathbf{x}_m^1)|}.$$

Thus the innovation is $\mathbf{v} = \mathbf{f}(\hat{\mathbf{x}}_l, \hat{\mathbf{x}}_m, \mathbf{d}) - \hat{\theta}$, with covariance $\mathbf{S} = \nabla\mathbf{f}\,\mathbf{Y}^{-1}\nabla^T\mathbf{f} + \mathbf{P}_\theta$, where $\nabla\mathbf{f}$ is the Jacobian of $\mathbf{f}(\cdot)$.

We form constraints between planes that associate and from this obtain a refined ML estimate of the joint map and vehicle trajectory. We refine the associations and state, iterating around 3 times or till convergence.

V. Experiment

We use a data set collected by the BAE Wildcat vehicle in the MIRA testing ground, shown in Fig. 7 with the prior map buildings highlighted. The trajectory is around 1.4 km long. The vehicle is equipped with an Oxford Technical Solutions INS system, differential GPS and a Velodyne laser scanner, which rotates at 10 Hz to produce a 360 degree 3D point cloud. The state consists of 77 poses, each with around 15,000 scan points.

The prior map consists of 477 planes representing vertical building walls, extracted from OS MasterMap GML data. We

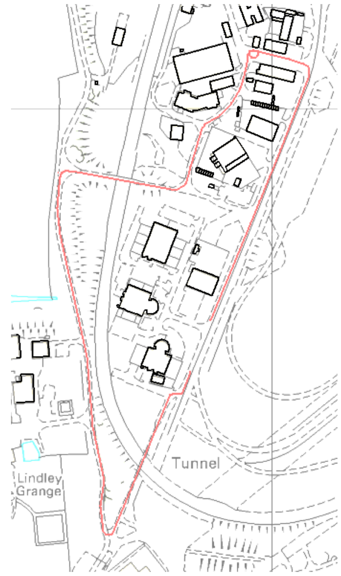


Fig. 7. Ordnance Survey map of the MIRA testing ground showing the prior map buildings and trajectory used in the experiment. The vehicle travels clockwise.

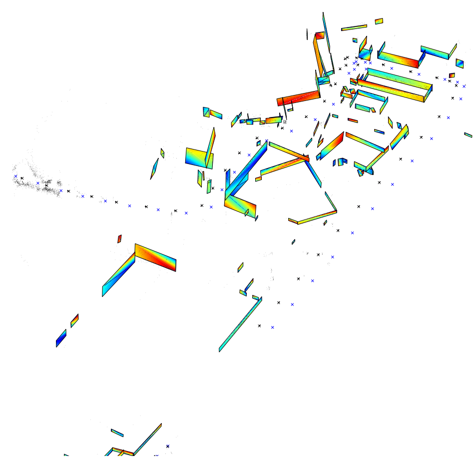


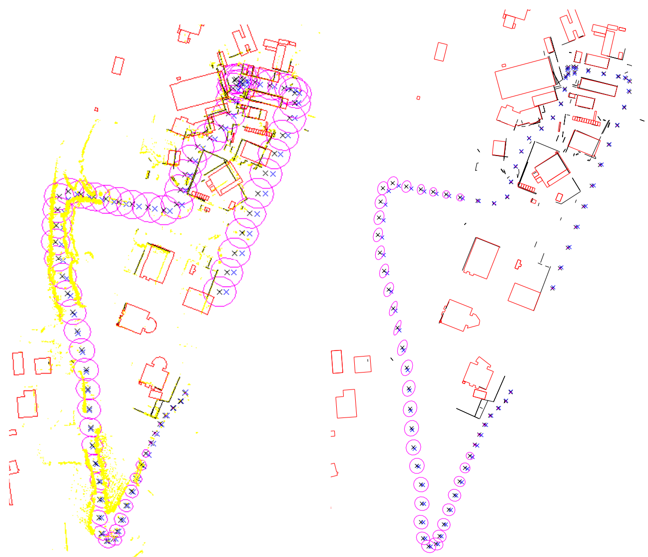
Fig. 8. Map showing the 125 planes extracted from the GraphSLAM map, overlaid on a small sample of the points they were derived from for comparison. Note how the sparsity of plane compared to point features means the map has a far smaller dimension, and how points associated with vegetation on the left have produced no planes.

assume that the information (precision) matrix of this map consists of two components for each vertex; an independent per-vertex uncertainty of 0.2 m (1σ), and a per-building uncertainty of 0.3 m (1σ), which is fully correlated between vertices belonging to the same building.

The ground truth vehicle position at each pose is obtained from GPS; these positions have an uncertainty of around 1 m (1σ).

Fig. 9(a) shows the trajectory and map of points produced after performing GraphSLAM, with the planes extracted from the point cloud overlaid on the prior map. Fig. 8 shows a 3D view of the planes extracted from the GraphSLAM map.

Fig. 9(b) shows the map and trajectory produced after



(a) Map before merging. (b) Map after merging.

Fig. 9. Overhead view of the trajectory and map before and after merging the prior information. Note how the map deformation has been corrected and the uncertainty in the entire trajectory has correspondingly reduced.

associating planes derived from the GraphSLAM map with those in the prior map. The ratio of the NEES [24] of the vehicle pose positions to the 99% χ^2 bound (with degree of freedom equal to the dimension of the positions) is 0.52 prior to and 0.6 post merging (accounting for the 1 m (1 σ) GPS uncertainty), indicating similar consistency, however the trace of the MSE has decreased from 3605 to 830 m^2, and the maximum position error from 13 to 8.7 m. This shows that the errors in the GraphSLAM state have been reduced by the information from the prior map, and the poses are closer to the ground truth.

VI. Conclusion

In this paper we presented a method of integrating a prior map into GraphSLAM by forming constraints directly between features in the state and prior map. Compared to methods which integrate maps into GraphSLAM indirectly through the platform poses, this approach allows for using prior maps with more general error characteristics, such as correlated errors and missing or spurious features. We validated the method on a data set consisting of a large loop in a built-up area.

In further work we plan to investigate the computational performance and perform experiments on a larger, more challenging dynamic environment.

Acknowledgements

This work was supported by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence (project AA018).

References

[1] 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.

[2] C. Brenner, "Towards fully automatic generation of city models," in *In: IAPRS*, 2000, pp. 85–92.

[3] 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.

[4] A. Harati and R. Siegwart, "Orthogonal 3D-SLAM for Indoor Environments Using Right Angle Corners," in *Proceedings of the 3rd European Conference on Mobile Robots (ECMR 2007)*, Freiburg, Germany, 19–21 September 2007, pp. 144–149.

[5] M. P. Parsley and S. J. Julier, "Towards the exploitation of prior information in slam," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 2991 –2996.

[6] F. Chanier, P. Checchin, C. Blanc, and L. Trassoudaine, "Map fusion based on a multi-map slam framework," Aug 2008, pp. 533–538.

[7] M. Parsley and S. Julier, "Slam with a heterogeneous prior map," in *Proceedings of the Fourth Annual SEAS DTC Conference*, 2009.

[8] K. W. Lee, S. Wijesoma, and J. I. Guzman, "A constrained slam approach to robust and accurate localisation of autonomous ground vehicles," *Robotics and Autonomous Systems*, vol. 55, no. 7, pp. 527–540, 2007.

[9] 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.

[10] S. Thrun and M. Montemerlo, "The graph slam algorithm with applications to large-scale mapping of urban structures," *Int. J. Rob. Res.*, vol. 25, no. 5-6, pp. 403–429, 2006.

[11] M. Karg, K. Wurm, C. Stachniss, K. Dietmayer, and W. Burgard, "Consistent mapping of multistory buildings by introducing global constraints to graph-based slam," in *Robotics and Automation, 2010. Proceedings. ICRA '10. IEEE International Conference on*, Anchorage, Alaska, 2010. [Online]. Available: http://www.informatik.uni-freiburg.de/ stachnis/pdf/karg10icra.pdf

[12] A. Trevor, J. Rogers, C. Nieto, and H. Christensen, "Applying domain knowledge to slam using virtual measurements," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may. 2010, pp. 5389 –5394.

[13] K. Takemura, A. Araki, J. Ido, Y. Matsumoto, J. Takamatsu, and T. Ogasawara, "Generating individual maps from universal map for heterogeneous mobile robots," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, may. 2010, pp. 3460–3465.

[14] J. Weingarten and R. Siegwart, "3d slam using planar segments," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 3062 –3067.

[15] A. Nüchter, 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.

[16] K. Lai and D. Fox, "Object recognition in 3d point clouds using web data and domain adaptation," *Int. J. Rob. Res.*, vol. 29, no. 8, pp. 1019–1037, 2010.

[17] 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.

[18] A. V. Segal, dirk Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: Science and Systems V*, 2009.

[19] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. on Robotics, TRO*, vol. 24, no. 6, pp. 1365–1378, Dec 2008.

[20] E. Olson, "Real-time correlative scan matching," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, June 2009, pp. 4387–4393.

[21] 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.

[22] G. Sibley, "A sliding window filter for slam," University of Southern California, Tech. Rep., 2006.

[23] T. A. Davis, *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2006.

[24] Y. Bar-Shalom and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2001.