Discovering Planes and Collapsing the State Space in Visual SLAM

Andrew P. Gee, Denis Chekhlov, Walterio Mayol and Andrew Calway

Department of Computer Science
University of Bristol, UK

{gee,chekhlov,wmayol,andrew}@cs.bris.ac.uk

Abstract

Recent advances in real-time visual SLAM have been based primarily on mapping isolated 3-D points. This presents difficulties when seeking to extend operation to wide areas, as the system state becomes large, requiring increasing computational effort. In this paper we present a novel approach to this problem in which planar structural components are embedded within the state to represent mapped points lying on a common plane. This collapses the state size, reducing computation and improving scalability, as well as giving a higher level scene description. Critically, the plane parameters are augmented into the SLAM state in a proper fashion, maintaining inherent uncertainties via a full covariance representation. Results for simulated data and for real-time operation demonstrate that the approach is effective.

1 Introduction

Concurrently estimating the pose of a moving camera and scene structure in real-time is appealing, both for the algorithmic challenges involved and the potential applications that can result. Recently, much progress has been made by adopting the principles of simultaneous localisation and mapping (SLAM) from robotics [19], where it was aimed at enabling autonomous systems to navigate around previously unknown surroundings. For a single camera, however, either attached to a person or handheld, and using only visual measurements, the problem is much more difficult, as the benefits of wheel odometry and control in robotics, for example, are no longer available.

Since the pioneering work of Davison [7], based on Kalman filtering and efficient feature matching, several visual SLAM systems have been proposed, addressing problems such as scalability and robustness. For example, Eade and Drummond [10] pose the problem within a factored sampling framework, using the FastSLAM algorithm to offer better theoretical scalability of map size, whilst Chekhlov et al. [5] use highly discriminative features with scale prediction to improve robustness of tracking and to handle periods of measurement loss. More recently, Williams et al. [20] combine an auxiliary RANSAC process with the Kalman filter to achieve relocalisation when tracking fails.

These systems are based on sparse maps, which are built by estimating the position of 3-D scene points and their associated uncertainties. This minimal representation is attractive since, in practice, it allows real-time operation whilst providing sufficient information to maintain accurate tracking. Recently, however, there has been a move towards the use of more structure-rich maps, in order to gain higher level scene descriptions and minimise redundancy. This includes the estimation of local surface normals [14], 3-D line segments [18, 12] and edgelets [9]. However, these descriptions are still based on isolated 3-D features and there is no sense of how these features group together to form higher level structures. A move towards the latter has been developed by Castle et al. [4], where known planar objects are detected using appearance features and inserted into the map.

In this paper we describe a novel approach to building in structural grouping within visual SLAM without prior knowledge of objects. This is achieved by augmenting the state with parameters representing higher level structures and then 'folding in' subsets of 3-D points which are consistent with the higher level parameterisation. This collapses the state space, reducing computational demands and increasing scalability, as well as giving a higher level scene description. Crucially, this is done in a manner which maintains the inherent uncertainties built up during SLAM operation by adopting a full covariance representation, ensuring consistency of the tracking and the map estimate. We demonstrate the approach by using it to discover 3-D planes in a scene. Plane detection has been previously investigated in the context of offline structure from motion [3] and is a natural choice to enhance visual maps given that most man-made environments contain large well defined planes. In Augmented Reality applications, for instance, planes are useful for augmenting the scene with virtual objects that are relative to a planar surface [13].

The paper is organised as follows. In the next section we briefly summarise the basic visual SLAM system. Details of state space augmentation and collapsing are then given in Section 3, followed by details of how this can be used for discovering higher level structure such as planar surfaces. Results are then presented for simulated data and real-time operation within an office environment.

2 Visual SLAM Using Kalman Filtering

We assume a calibrated camera moving with agile motion whilst viewing a static scene. The aim is to estimate the camera pose, whilst simultaneously estimating the 3-D parameters of structural features, such as points or surfaces. We use a Kalman filter framework, similar to that in [7, 18, 5]. The system state $\mathbf{x} = [\mathbf{v}, \mathbf{m}]^T$ has two partitions, one for the camera pose $\mathbf{v} = [\mathbf{q}, \mathbf{t}]^T$ and one for the structural features, $\mathbf{m} = [\mathbf{m}_1, \mathbf{m}_2, \ldots]^T$. The camera rotation is represented by the quaternion $\mathbf{q}$ and $\mathbf{t}$ is its 3-D position vector, both defined w.r.t a world coordinate system. We adopt a generalised framework for the structural partition to allow multiple feature types to be mapped. Each $\mathbf{m}_i$ denotes a parameterisation of a feature, be it a position vector for a 3-D point or the position and orientation of a planar surface. In general, we expect the number and the types of features to vary as the camera explores the environment, with new features being added and old ones being removed so as to minimise computation and maintain stability.

The Kalman filter requires a *process model* and an *observation model*, encoding the assumed evolution of the state between time steps and the relationship between the state and the filter measurements, respectively [2]. We assume a constant position model for the camera motion, i.e. a random walk, and since we are dealing with a static scene this gives the process model

$$\mathbf{x}^{new} = \mathbf{f}(\mathbf{x}, \mathbf{w}) = [\Delta\mathbf{q}(\mathbf{w}_\omega) \otimes \mathbf{q}, \mathbf{t} + \mathbf{w}_\tau, \mathbf{m}]^T \tag{1}$$

where $\mathbf{w} = [\mathbf{w}_\omega, \mathbf{w}_\tau]^T$ is a 6-D noise vector, assumed to be from $\mathbf{N}(0, \mathbf{Q})$, $\Delta\mathbf{q}(\mathbf{w}_\omega)$ is the incremental quaternion corresponding to the Euler angles defined by $\mathbf{w}_\omega$, and $\otimes$ denotes quaternion multiplication. Note that the map parameters are predicted to remain unchanged and that the non-additive noise component in the quaternion part is necessary to give an unbiased distribution in rotation space. At each time step, we collect a set of measurements, $\mathbf{z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots]^T$, where in general each $\mathbf{z}_i$ will be related to the state via a separate measurement function, i.e. $\mathbf{z}_i = \mathbf{h}_i(\mathbf{x}, \mathbf{e})$, where $\mathbf{e}$ is a multivariate noise vector from $\mathbf{N}(0, \mathbf{R})$. For example, when mapping point features, the measurements are assumed to be corrupted 2-D positions of projected points and for the *i*th measurement

$$\mathbf{h}_i(\mathbf{x}, \mathbf{e}) = \Pi(\mathbf{y}(\mathbf{v}, \mathbf{m}_j)) + \mathbf{e}_i \tag{2}$$

where $\mathbf{m}_j$ is the 3-D position vector of the point associated with the measurement $\mathbf{z}_i$, $\mathbf{y}(\mathbf{v}, \mathbf{m}_j)$ denotes this position vector in the camera coordinate system, Π denotes pin-hole projection for a calibrated camera and $\mathbf{e}_i$ is a 2-D noise vector from $\mathbf{N}(0, \mathbf{R}_i)$.

The filter gives state mean and covariance estimates based on the process and observation models. Both are non-linear and thus we use the extended Kalman filter, in which predictions and updates are derived from approximations based on the Jacobians of $\mathbf{f}$ and $\mathbf{h}_i$ [2]. The role of the covariances within the filter is critical. Proper maintenance ensures that updates are propagated amongst the state elements, particularly the structural components, which are correlated through their mutual dependence on the camera pose [19, 8]. This ensures consistency and stability of the filter. Propagation of the state covariance through the observation model also allows for better data association, constraining the search for image features. As discussed next, it is therefore crucial to ensure that covariances are properly maintained for successful filter operation.

3 EKF SLAM with Plane Discovery

We discover higher level structure within the SLAM framework in the following manner. We begin by mapping 3-D points, starting from known points on a calibration pattern [7]. As the camera moves away, new points are initialised and added to the map, allowing tracking to continue as the camera explores the scene. Points are added using the inverse depth representation [15] and augmented to the state in a manner which maintains full covariance with the rest of the map [1] (see below). For measurements, we use the fast salient point detector developed by Rosten and Drummond [17] to identify potential features and the SIFT-like descriptors with scale prediction developed by Chekhlov *et al.* [5] for repeatable matching across frames.

As SLAM proceeds, the system seeks to identify subsets of mapped points that potentially lie on a planar structure. This is done using an auxiliary RANSAC process and a subsequent principal component analysis amongst inliers then yields initial estimates for the plane parameters, allowing a new ‘planar feature’ to be augmented to the state. The 3-D inlier points are then transformed to 2-D planar points, representing their position within the plane, hence collapsing the state. Critically, both the augmentation and transformation include proper adjustment of the state covariance, ensuring that full cross correlation is maintained amongst the existing and new state parameters. As SLAM continues, new 3-D planes are initialised as appropriate and the system attempts to collapse new 3-D points into the planes, hence continuing to reduce state size. Details of each component are given in the following sections.

3.1 Augmenting and Collapsing the State Space

It is useful to consider first the general procedure for augmenting and transforming the state space in a SLAM system [1, 6]. Assume that we have an existing state estimate $\hat{\mathbf{x}} = [\hat{\mathbf{v}}, \hat{\mathbf{m}}_1, \ldots, \hat{\mathbf{m}}_n]^T$, with n features in the map, and that we wish to augment the state with a new feature $\mathbf{m}_{n+1}$, based on an initialisation measurement $\mathbf{z}_o$. In general, the initial estimate for the feature will be derived from a combination of the measurement and the existing state, i.e. $\hat{\mathbf{m}}_{n+1} = \mathbf{s}(\hat{\mathbf{x}}, \mathbf{z}_o)$, and the augmented state covariance then becomes

$$\mathbf{P}^{new} = \mathbf{J} \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_o \end{bmatrix} \mathbf{J}^T \qquad \mathbf{J} = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{0} \\ \hline \nabla \mathbf{s}_{\mathbf{v}} \ \nabla \mathbf{s}_{\mathbf{m}_1} \ \ldots \ \nabla \mathbf{s}_{\mathbf{m}_n} & \nabla \mathbf{s}_{\mathbf{z}_o} \end{array} \right] \tag{3}$$

where $\mathbf{R}_o$ is the covariance of the measurement and $\nabla \mathbf{s}_{\mathbf{v}} = \partial \mathbf{s} / \partial \mathbf{v}$. This transformation of the covariance introduces the important correlations between the new feature and the existing features in the map. For example, in the case of augmentation with a new point feature, only the Jacobians $\nabla \mathbf{s}_{\mathbf{v}}$ and $\nabla \mathbf{s}_{\mathbf{z}_o}$ are non-zero and the new point feature is correlated with those already in the map through the camera pose. In a similar way, we can also transform an existing feature to a different representation, e.g. $\hat{\mathbf{m}}_i^{new} = \mathbf{r}(\hat{\mathbf{x}})$, and then the covariance update is given by

$$\mathbf{P}^{new} = \mathbf{J}\mathbf{P}\mathbf{J}^T \qquad \mathbf{J} = \left[\begin{array}{c|c|c} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline \nabla \mathbf{r}_{\mathbf{v}} \ \ldots \ \nabla \mathbf{r}_{\mathbf{m}_{i-1}} & \nabla \mathbf{r}_{\mathbf{m}_i} & \nabla \mathbf{r}_{\mathbf{m}_{i+1}} \ \ldots \ \nabla \mathbf{r}_{\mathbf{m}_n} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{I} \end{array} \right] \tag{4}$$

where any change in the dimensions of the state is reflected in the dimensions of the Jacobian $\mathbf{J}$. We can use both of the above relationships in order to introduce higher level features and collapse the state size. For example, consider a new feature $\mathbf{m}_{n+1}$ which represents a higher level structure and which imposes a constraint on a subset of existing features. Given an initial estimate of $\mathbf{m}_{n+1}$, we can introduce it into the system using (3) and the constraint can then be imposed by transforming the existing features into the new constrained representation using (4). This collapses the state by removing redundant constrained variables, whilst also maintaining the correlations between the new features, the camera pose and the remainder of the map.

3.2 Representing Planes

We demonstrate the process of collapsing the state space using the example of discovering and enforcing planar structure in a scene. The descriptors used to match and identify 3-D points perform best when they are initialised on planar, textured surfaces, so it is likely that many of the points in the map will end up lying on planes. We seek to infer the locations and orientations of these planes and introduce parameterisations of the planes as new features in the map. This then allows the 3-D points lying on the planes to be collapsed into 2-D planar points defined w.r.t the plane. We use a seven parameter state vector to define a plane, so that a new planar feature augmented to the state has the form $\mathbf{m}_{n+1} = [\mathbf{p}_o, \theta_1, \phi_1, \theta_2, \phi_2]^T$, where $\mathbf{p}_o$ is the plane origin and the orientation is defined by two basis vectors, $\mathbf{c}(\theta_1, \phi_1)$ and $\mathbf{c}(\theta_2, \phi_2)$, which lie on the plane, i.e.

$$\mathbf{c}(\theta_i, \phi_i) = [cos\phi_i \ sin\theta_i, \ -sin\phi_i, \ cos\phi_i \ cos\theta_i]^T \tag{5}$$

Note that the normal to the plane is then simply the cross product between the two basis vectors, i.e. $\mathbf{n} = \mathbf{c}(\theta_1, \phi_1) \times \mathbf{c}(\theta_2, \phi_2)$. A 3-D point in the map which lies on the plane and whose feature vector $\mathbf{m}_i$ defines its 3-D position vector, can then be transformed into a 2-D planar point using

$$\mathbf{m}_i^{new} = [(\mathbf{m}_i - \mathbf{p}_o) \cdot \mathbf{c}(\theta_1, \phi_1), \ (\mathbf{m}_i - \mathbf{p}_o) \cdot \mathbf{c}(\theta_2, \phi_2)]^T \tag{6}$$

where $\cdot$ denotes the dot product. Thus for l such points, this gives a state size of $7 + 2l$, compared with $3l$, giving a reduction in state size for $l > 7$.

3.3 Initialising Planes

We use the RANSAC algorithm [11] to search for planes amongst point features in the map, similar in approach to that used in [3]. Plane hypotheses are generated from minimal sets of points randomly sampled from the subset of point features with variance $\sigma_{max}^2 < \sigma_T^2$, where σ_{max}^2 is the maximum of the variances along each dimension and σ_T is a suitably chosen threshold. Each hypothesis is tested for consensus with the rest of the set. A minimal set of point features $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ generates the parameters of the plane hypothesis as follows:

$$\mathbf{p}_o = \mathbf{m}_1 \qquad \mathbf{c}(\theta_1, \phi_1) = \mathbf{m}_2 - \mathbf{m}_1 \qquad \mathbf{c}(\theta_2, \phi_2) = \mathbf{m}_3 - \mathbf{m}_1 \tag{7}$$

and a point $\mathbf{m}_i$ is deemed to be in consensus with the hypothesis if its perpendicular distance from the plane, d, is less than d_T, where $d = (\mathbf{m}_i - \mathbf{p}_o) \cdot \mathbf{n}$, and its Euclidean distance from the plane origin is less than d_{max}. The second test ensures that we only initialise planes with strong local support. This is important because it is known that the relative positions of nearby points in a SLAM system are typically very accurate, even if the uncertainty in the global positions is large, so introducing a plane using a set of local points is relatively safe. However, we can make no such assumptions about the relative positions of widely dispersed points in the SLAM map.

The best-fit plane is determined from the inlying point features for the plane hypothesis with most consensus. The origin is set to the mean and the orientation parameters are determined from the principal components. Specifically, if the position vectors for l inlying points w.r.t the mean are stacked into an $l \times 3$ matrix $\mathbf{M}$, then the eigenvector of $\mathbf{M}^T\mathbf{M}$ corresponding to the smallest eigenvalue gives the normal to the plane and the other two eigenvectors give the basis vectors within the plane. The smallest eigenvalue, λ_{min}, is the variance of the inliers in the normal direction and provides a convenient measure of the quality of the fit.

In order to avoid adding poor estimates of planes to the system state, the best-fit plane generated by the RANSAC process is only initialised in the SLAM system if $l > l_T$ and $\lambda_{min} < \lambda_T$. The best-fit parameters are used to initialise a plane feature in the state and the covariance is updated according to (3), where $\mathbf{s}(\mathbf{x}, \mathbf{z}_o)$ denotes the derivation of the plane parameters from the set of inlying point features. Thus, both $\mathbf{z}_o = \mathbf{0}$ and $\nabla\mathbf{s}_\mathbf{v} = \mathbf{0}$, whilst the Jacobian w.r.t an inlying point feature $\mathbf{m}_i$ is computed as:

$$\nabla\mathbf{s}_{\mathbf{m}_i} = \left[\frac{\partial \mathbf{p}_o}{\partial \mathbf{m}_i} \ , \ \frac{\partial(\theta_1, \phi_1)}{\partial \mathbf{c}(\theta_1, \phi_1)} \frac{\partial \mathbf{c}(\theta_1, \phi_1)}{\partial \mathbf{m}_i} \ , \ \frac{\partial(\theta_2, \phi_2)}{\partial \mathbf{c}(\theta_2, \phi_2)} \frac{\partial \mathbf{c}(\theta_2, \phi_2)}{\partial \mathbf{m}_i} \right]^T \tag{8}$$

where $\partial\mathbf{c}(\theta_1, \phi_1)/\partial\mathbf{m}_i$ and $\partial\mathbf{c}(\theta_2, \phi_2)/\partial\mathbf{m}_j$ are the Jacobians of the two eigenvectors computed using the method described in [16]. This ensures that the plane parameters are correlated with the rest of the SLAM state through the inlying point features. However, we do not immediately convert these inlying feature points to 2-D points on the plane in case the latter proves to be an unstable addition to the map. Instead, we choose to gradually add points to the plane using the mechanism described in the next section.

3.4 Adding and Fixing Points in Planes

At each time step, all converged 3-D point features in the map are considered as candidates for transformation to 2-D points associated with planes augmented to the map. A 3-D point feature is transformed using (6) if $\sigma_{max}^2 < \sigma_T^2$ and its perpendicular distance from the plane and its distance from the plane origin are within d_T and d_{max}, respectively. The state covariance is then updated using (4). Since the new representation is dependent on the plane parameters, this process introduces correlations with the other 2-D points associated with the same plane.

If the maximum of the variances of a 2-D planar point becomes very small, less than a threshold σ_{fix}, then we can consider a further collapse of the state space by removing it completely. Instead of maintaining an estimate of its 2-D position in the plane, we consider it as a fixed point in the plane and use its measurements to update the associated planar feature directly (as described in the next section). This leads to a significant reduction in the size of the state space but does so at the cost of introducing errors and inconsistency into the system (c.f. Section 4).

3.5 Plane Measurement Equation

It is not possible to make a direct observation of the plane. However, each observed feature on the plane imposes a constraint between the camera pose, the position and orientation of the plane and the position of the observed 2-D point. The measurement model for a 2-D planar point $\mathbf{m}_i$ is very similar to that of a standard 3-D point feature, but involves an additional preliminary step to convert it to a 3-D position vector $\mathbf{p}$ in the world frame of reference prior to projection into the camera, i.e. from (6)

$$\mathbf{p} = [\mathbf{c}(\theta_1, \phi_1) \;\; \mathbf{c}(\theta_2, \phi_2)]\, \mathbf{m}_i \, + \, \mathbf{p}_o \tag{9}$$

The predicted measurement for the planar point can then be obtained by passing $\mathbf{p}$ through the standard perspective projection model in (2). The similarity between the measurement models makes the implementation of planar points in the existing EKF SLAM system very simple. The measurement Jacobian required by the EKF is much the same as that for 3-D point features, except that we need to take account of the conversion in (9) when computing the Jacobian relating the observation to 2-D point feature, i.e

$$\frac{\partial \mathbf{h}_i}{\partial \mathbf{m}_i} = \frac{\partial \mathbf{h}_i}{\partial \mathbf{p}} \frac{\partial \mathbf{p}}{\partial \mathbf{m}_i} \tag{10}$$

where $\mathbf{h}_i$ is the measurement function associated with a 3-D point (eqn 2) and $\partial \mathbf{p}/\partial \mathbf{m}_i$ is the Jacobian of the 3-D point w.r.t the 2-D planar point, derived from (9). Finally, since the predicted observation is also dependent on the position and orientation of the plane, a Jacobian $\partial \mathbf{h}_i/\partial \mathbf{m}_j$ relating $\mathbf{h}_i$ to the relevant plane feature $\mathbf{m}_j$ is also derived from (9).

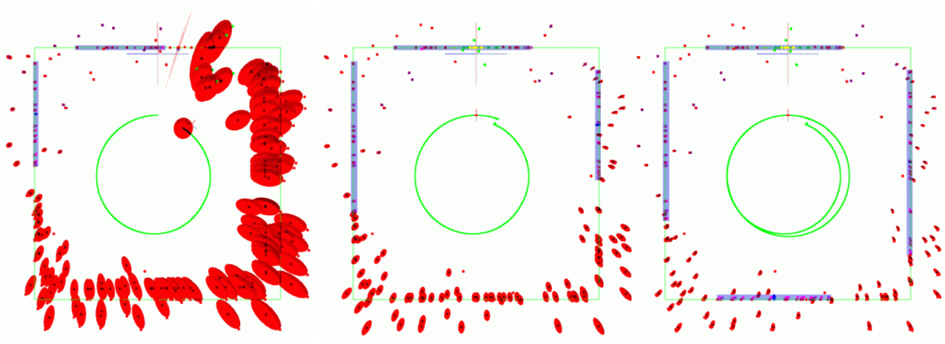

Figure 1: Simulation results: (left) large covariance just prior to loop closure; (middle) covariance and map error reduction after loop closure; (right) further reductions in covariance and discovery of additional plane on second loop.

4 Experiments

Experiments were carried out to test the effectiveness of plane discovery and the effect of collapsing the state space on the consistency of the SLAM estimation. Simulated data was used to determine the accuracy of tracking and reconstruction against a known ground truth. Performance was also assessed for real-time operation using an agile hand-held camera within an office environment.

The simulation models a 6 d.o.f. camera moving through an environment containing 200 3-D points arranged in a square room with walls 4m long (Fig. 1) and with 50% of the points lying at random positions on the walls. Newly observed points are initialised as 6-D inverse depth points and converted to 3-D points once their depth uncertainty becomes Gaussian [6]. Camera specifications are 43° horizontal field of view (FOV) and image size 320×240 pixels. Point features are observed with perfect data association and zero-mean Gaussian measurement error of 1.0 pixel^2 variance. The camera moves on a fixed circular trajectory of radius 1m around the room centre and maintains a radial orientation. Each simulation was run for two full loops of the camera trajectory. We used the following thresholds, chosen to minimise mismatches of points to planes: $d_T = 0.5cm$, $l_T = 7$, $\lambda_T = d_T^2$, $d_{max} = 200cm$. Varying these values produces different numbers and sizes of planes. The simulations were run for three different SLAM scenarios: with points only; with inference of planes; and with inference of planes and fixed planar points. Different values for the thresholds on point feature variance were tested: a strong threshold $\sigma_T = 2.0cm$, $\sigma_{fix} = 0.1cm$ (cases B and D in Fig. 2); a weak threshold $\sigma_T = 10.0cm$, $\sigma_{fix} = 1.0cm$ (cases C and E); and a medium threshold $\sigma_T = 10.0cm$, $\sigma_{fix} = 0.1cm$ (case F).

Figure 1 shows an external view of the camera position, trajectory, and map estimates prior to and following loop closure. The ellipses indicate position and point variances and the plane estimates are shown superimposed on the ground-truth outline of the room. In this case, a strong threshold was used for plane discovery and planar points were not fixed. Note the accuracy of the plane estimates and the clear reduction in point covariance and correction of the camera trajectory after loop closure. This demonstrates that inference of higher level structure and collapsing of the state space can be achieved without losing consistency. Reductions in state size for the different simulations are shown in

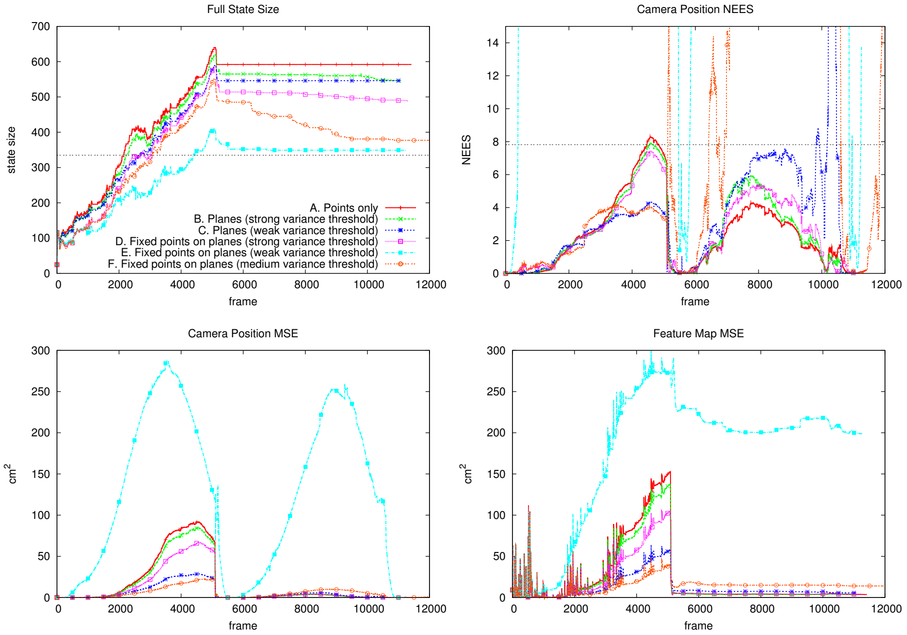

Figure 2: Simulation results: (top-left) total state size, horizontal line indicates minimum size if all planar points are added as fixed points; (top-right) NEES of camera position, horizontal line indicates consistency threshold; (bottom-left) MSE of camera position; (bottom-right) MSE of map features. Loop closure occurs in frames 5400 and 10800.

Fig. 2 (top-left). We observe that the different methods all lead to different levels of state reduction and the merits of each approach can be assessed by comparing the consistency and accuracy shown in the remaining plots of Fig. 2. Consider case E which achieves a large reduction in state size but becomes inconsistent and with large mean-squared error (MSE). In contrast, case D achieves savings in state size compared to points only (case A) but maintains consistency. The consistency of the camera position estimates (Fig. 2 (top-right)) was measured using the Normalised Estimation Error Squared (NEES) [2] function $(\mathbf{v}-\hat{\mathbf{v}})^{\mathsf{T}}\hat{\mathbf{P}}_{\mathbf{vv}}^{-1}(\mathbf{v}-\hat{\mathbf{v}})$. The figure also shows the threshold $\chi^2_{r,1-\alpha}$ from the χ^2 distribution with $r = dim(\mathbf{v}) = 7$ degrees of freedom and α the desired significance level (0.95 in our experiments). Note that consistency is strongly affected when we try to aggressively fix planar points to reduce state size (cases C, E and F). However, more conservative approaches (cases B and D) achieve useful state space reductions and introduce higher order structure without adversely affecting consistency.

MSE analysis of camera and feature position estimates (Fig. 2 (bottom)) shows the cost of introducing inconsistency. The most inconsistent methods do not benefit fully from loop closure and are unable to correct the error in their maps. More conservative thresholds maintain consistency and achieve similar accuracy to the points only approach with the additional benefit of reduced state size. There is also an indication that the MSE during the first loop is reduced when we add planes to the system (cases B, C, D and F). This may be because a good plane estimate extends its influence further into the map than a good point, due to its higher level structure, and any points constrained to lie on

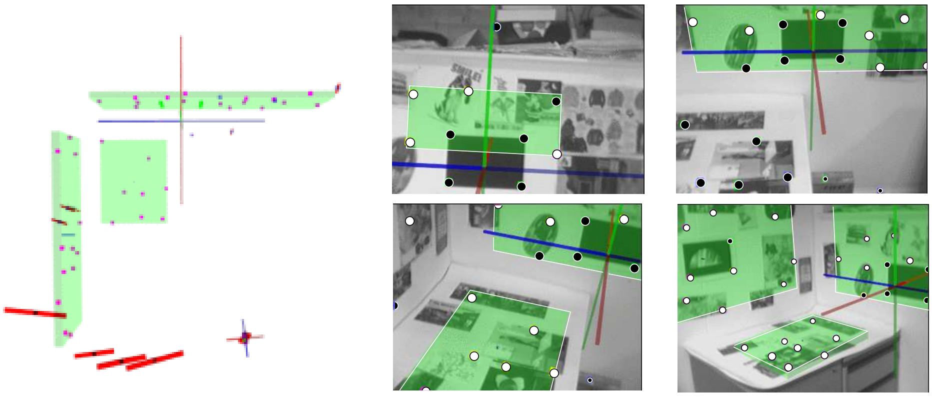

Figure 3: Real-time operation with hand-held camera: (left) view of camera position, map estimates and planar features; (middle and right) planar features (boundaries in white) are discovered with converged 3-D points in black and 2-D planar points in white. Videos available at this paper's entry at `www.cs.bris.ac.uk/Publications`.

this plane will also have their accuracy improved. This effect could be exploited by using known planes to initialise the SLAM system.

We also tested the method in real-time in an office environment using a calibrated hand-held web-cam with a resolution of 320×240 pixels and 43° FOV. Tracking was initialised with four known map points corresponding to the corners of a planar black on white rectangle placed in the scene. The environment contained three distinct planar surfaces with sufficient texture to encourage reliable feature detection and recognition. The former was based on a fast salient operator [17] and the latter on multiscale SIFT-like descriptors with scale prediction [5]. This gives robust performance, even in the face of significant erratic camera movement. Figure 3 shows an external view of the camera position and map estimates part way through the sequence and views through the camera with mapped point and planar features superimposed. Note that the three planes in the scene have been successfully detected and the vast majority of the 3-D points mapped were transformed into 2-D planar points (except those on the calibration pattern which for fair comparison we force to remain as 3-D points). In this experiment, the system operated at around 20 frames/sec, including graphics rendering and without software optimsation, for maps of up to 50 features and with 10 visible features per frame on average.

5 Conclusions

Augmenting the SLAM state with parameters representing a planar structure allows us to represent mapped points lying on the plane with a reduced parameterisation and collapse the state space whilst maintaining a consistent, full-covariance representation. This technique can be applied in real-time visual SLAM systems to reduce computational demands, as well as giving a higher level scene description suitable for further applications. Future work will consider the incorporation of a wider variety of structural information, such as lines and junctions. Knowledge of planar structure in the scene may also have benefits for visual feature detection and matching which can be exploited to improve performance.

Acknowledgements

This work was funded by the EPSRC Equator IRC and ORSAS UK.

References

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping (slam): Part ii - state of the art. *IEEE Robotics and Automation Magazine*, (3), 2006.

[2] Y. Bar-Shalom, T. Kirubarajan, and X.R Li. *Estimation with Applications to Tracking and Navigation*. 2002.

[3] A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105(1):42–59, 2007.

[4] R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proc. Int. Conf. Robotics and Automation*, 2007.

[5] D. Chekhlov, M. Pupilli, W. Mayol, and A. Calway. Robust real-time visual slam using scale prediction and exemplar based feature description. In *Proc. Int Conf on Computer Vision and Pattern Recognition*, 2007.

[6] J. Civera, A.J. Davison, and J.M.M. Montiel. Inverse depth to depth conversion for monocular slam. In *Proc. Int. Conf. Robotics and Automation*, 2007.

[7] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. Int. Conf. on Computer Vision*, 2003.

[8] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (slam): Part i the essential algorithms. *IEEE Robotics and Automation Magazine*, (2), 2006.

[9] E. Eade and T. Drummond. Edge landmarks in monocular slam. In *Proc. British Machine Vision Conf*, 2006.

[10] E. Eade and T. Drummond. Scalable monocular slam. In *Proc. Int Conf on Computer Vision and Pattern Recognition*, 2006.

[11] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[12] A.P. Gee and W. Mayol-Cuevas. Real-time model-based slam using line segments. In *Int. Symp. on Visual Computing*, 2006.

[13] W.W. Mayol, A.J. Davison, B.J. Tordoff, N.D. Molton, and D.W. Murray. Interaction between hand and wearable camera in 2d and 3d environments. In *Proc. British Machine Vision Conf*, 2004.

[14] N. Molton, I. Ried, and A.J Davison. Locally planar patch features for real-time structure from motion. In *Proc. British Machine Vision Conf*, 2004.

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

[16] T. Papadopoulo and M.I.A. Lourakis. Estimating the jacobian of the singular value decomposition: Theory and applications. In *Proc. Euro Conf on Computer Vision*, 2000.

[17] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proc. Euro Conf on Computer Vision*, 2006.

[18] P. Smith, I. Reid, and A.J. Davison. Real-time monocular slam with straight lines. In *Proc. British Machine Vision Conf.*, 2006.

[19] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In *Proc. Int. Symp. Robotics Research*, 1987.

[20] B. Williams, P. Smith, and I. Reid. Automatic relocalisation for a single-camera simultaneous localisation and mapping system. In *Proc. Int. Conf. on Robotics and Automation*, 2007.