Locally Planar Patch Features for Real-Time Structure from Motion

Nicholas Molton, Andrew Davison and Ian Reid
Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK
[ndm,ajd,ian]@robots.ox.ac.uk

Abstract

The performance of sequential structure from motion systems, where scene mapping is sparse to permit real-time operation, depends greatly on the ability to repeatedly measure the same visual features from a wide range of viewpoints. While previous systems have tracked features as 2D templates in image space, we show that long-term tracking is improved by treating salient feature patches as observations of locally planar regions on 3D world surfaces. Within a SLAM framework for motion and structure estimation, a gradient-based image alignment method is used to deduce estimates feature surface normal estimates, enabling pre-warping of templates for matching. As an added benefit these normals provide a richer description of the scene.

1 Introduction

Real-time structure from motion systems are relatively rare, arriving on the scene recently as PC performance has caught up with their processing requirements. Such systems have a wide range of potential applications because they enable a standard camera with no extra equipment to serve as a flexible real-time position sensor, useful in robotics, wearable technology and augmented reality. Some have presented real-time algorithms which string together frame-to-frame motion and structure estimates achieved in the well-known structure from motion style [8]. While demonstrating impressive performance in local motion recovery, however, such methods cannot be used for repeatable localisation during long periods because they are unable to re-register with parts of the scene seen earlier.

Repeatable localisation requires a persistent map of feature landmarks. The process of building such a map on-line while at the same time estimating sensor motion is known as Simultaneous Localisation and Mapping (SLAM). Bayesian SLAM algorithms based on the Extended Kalman Filter (EKF) are in wide use in the mobile robotics community, but generally using controlled robot platforms with non-visual sensors. In this paper we build on a recent step forward demonstrating real-time SLAM using only a single camera, freely moving in 3D [2] — bringing the SLAM methodology into the "pure vision" arena.

1.1 Single Camera SLAM and and Persistent Landmarks

In the single camera SLAM approach of [2], estimates of the locations of a calibrated camera and the arbitrary point features it observes are stored probabilistically in a single

BMVC 2004 doi:10.5244/C.18.90

state vector $\mathbf{x}$ and covariance matrix $\mathtt{P}$:

$$\hat{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}}_v \\ \hat{\mathbf{y}}_1 \\ \hat{\mathbf{y}}_2 \\ \vdots \end{pmatrix}, \ \mathtt{P} = \begin{bmatrix} \mathtt{P}_{xx} & \mathtt{P}_{xy_1} & \mathtt{P}_{xy_2} & \dots \\ \mathtt{P}_{y_1x} & \mathtt{P}_{y_1y_1} & \mathtt{P}_{y_1y_2} & \dots \\ \mathtt{P}_{y_2x} & \mathtt{P}_{y_2y_1} & \mathtt{P}_{y_2y_2} & \dots \\ \vdots & \vdots & \vdots & \end{bmatrix}. \tag{1}$$

Here the camera state estimate $\mathbf{x}_v$ comprises a metric 3D position vector $\mathbf{r}^W$, orientation quaternion $\mathbf{q}^{RW}$, velocity vector $\mathbf{v}^W$ and angular velocity vector $\boldsymbol{\omega}^W$ relative to a fixed world frame. Feature states $\mathbf{y}_i$ are 3D position vectors comprising the scene "map".

These estimates are updated sequentially in an Extended Kalman Filter to account for: 1) camera movement between image aquisitions (at 30Hz); and 2) re-measurement of mapped features. Feature states can be dynamically added to the map when new landmarks are required, and can also be deleted if map management criteria deem this necessary. Storing this information in a single state vector and covariance is important because camera and feature location estimates become correlated during these updates.

The map is sparse to permit real-time processing, and it is therefore critical that the small set of features used act as long-term, high-quality landmarks — this is what will prevent the drift of motion estimates with time. In the system of [2], feature matching is on the basis of simple 2D image templates, taking no account of the change in feature appearance as the viewpoint changes — the result is a severely limited range of camera movement within which each landmark is visible, and therefore a requirement to add extra features to the map at the cost of efficiency and increased map uncertainty. Approaches other authors have taken include the natural step of re-initialising a feature's template once the viewpoint has changed significantly. Se et al [9] used this method in their SLAM system based on scale and rotation-invariant SIFT features to further increase the range of feature visibility. Updating templates is however in general undesirable because it is not longer certain that the feature represents a unique 3D world point.

1.2 Features as Locally Planar 3D Patches

In this paper we aim to make full use of the advantages of a sequential SLAM system to enable robust wide-baseline matching of features without updating their templates. The key assumption is to assert that each salient image patch detected as a feature candidate (using the Shi and Tomasi operator as in [2]) corresponds to an observation of a locally planar surface in the 3D scene, rather than a 2D image entity. This approximation of flatness is relative to the size of camera motion over which the feature will be observed, but many features in typical scenes fit this assumption over usual movement ranges — particularly in the case common in localisation within a room-like concave space. The salient image texture is attached to this 3D surface, and then when a feature measurement is required the current estimates of camera and feature position from the SLAM state vector can be used to pre-warp the texture to the appearance expected from this viewpoint.

Calculating these warps requires an estimate not just of the patch's 3D position (obtained as usual in the SLAM algorithm), but also its surface normal. A key contribution of this paper is to present a method for normal estimation using gradient-based image alignment and demonstrate its full integration in a real-time SLAM system at 30Hz.

2 Estimating Surface Normal through Image Alignment

In our system, when a new feature is first detected and added to the SLAM map a region from the initial image is saved as a 2D template. Now that we assume that a feature corresponds to a locally planar region in 3D space, its image appearance will be transformed by changes in viewpoint by warping this initial template in a way that can always be characterised by a 2D homography. The exact nature of the warp (which may include scaling, rotation, shearing and more general projective warping) depends on the initial and current positions of the camera, the 3D position of the centre of the feature, and the orientation of its local surface. The SLAM system provides a running estimate of camera pose and 3D feature positions, and we now additionally maintain estimates of the initial camera position and the local surface orientation for each point. This allows a *prediction* of the feature's appearance from the current viewpoint to be made before it is measured, and then an *update* of the surface orientation estimate to be made after measurement.

It is assumed that this prediction of appearance is sufficient for the current image position of the feature to be found using a standard exhaustive image correlation search within an elliptical uncertainty region derived from the SLAM filter. However, the small difference between the pre-warped template and the feature's current appearance provides additional information on how the true surface normal differs from the current estimate. Gradient-descent image alignment is therefore applied to these two image regions to correct the warp and provide an improved estimate of the surface normal.

Immediately after detecting a new feature as a salient image region we initialise an estimate of its surface normal which is parallel to the current viewing direction, but this estimate is highly uncertain (we might go so far as to say it is completely unknown, but we do have some prior information because the fact that we have seen this surface at all tells us that it is more likely to be facing the camera than at a very acute angle). While the camera is close to this intitialisation position however, the patch's appearance in the current image is highly insensitive the precise surface normal (this is the reason why 2D image matching works in most short-baseline stereo or structure from motion systems). This means that will be able to successfully match the feature without difficulty during these first motions, but also conversely that only weak information is available to enable deduction of the actual normal. Nevertheless, by the time the camera has moved enough for the surface normal estimate to affect the feature's appearance significantly we hope to have repeatedly made small improvements to our estimate of surface normal.

We currently make the simplifying approximation that estimates of feature normals are only weakly correlated to those of camera and feature positions. Normal estimates are therefore not stored in the main SLAM state vector, but maintained separately. This means that warp measurements cannot directly be used to provide information on camera motion (in the sense that a patch whose scale increases indicates motion towards that feature, and so on). Warping and normal estimation is used only as an aid to robust feature matching and it is the image positions of the matches which are used to update the SLAM filter.

Related work includes that of Jin *et al* [4], whose sequential structure from motion algorithm estimates camera pose and the position and orientation of locally planar features directly by comparison of the current images to that implied by the estimated system state. This method relies on the linearity of image gradient, which limits the system to slow camera motions, and the method is not described as working in real time. Our approach is also related to the stereo work of Hattori *et al* [3], which performs stereo matching under

an affine warp parameterised in terms of feature depth and surface orientation.

2.1 The Transformation of a Surface between Two Views

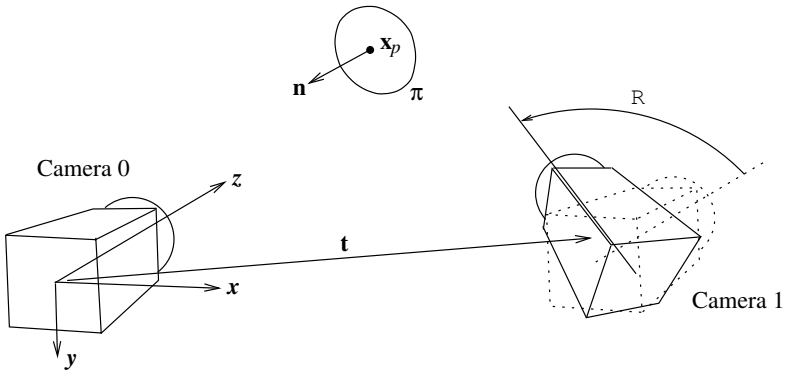

Figure 1: Parameters describing the geometry of two cameras and a locally planar surface.

We first analyse the geometry of two cameras viewing a planar patch. Figure 1 shows a scene containing a camera (Camera 0) at the origin with zero rotation, a second camera (Camera 1) at position $\mathbf{t}$, rotated by $\mathtt{R}$ from the first, and a point at position $\mathbf{x}_p$, which is on a locally planar surface with normal $\mathbf{n}$. A point at position (u_0, v_0) in the image plane of camera 0 backprojects to the ray $\rho\mathbf{u}_0$, where $\mathbf{u}_0 = (u_0, v_0, 1)^T$. Points $\mathbf{x}$ on the plane π satisfy the equation $\mathbf{n}^T\mathbf{x} = \mathbf{n}^T\mathbf{x}_p$. At the intersection of the ray with the plane, $\mathbf{x} = \rho\mathbf{u}_0$, so $\rho\mathbf{n}^T\mathbf{u}_0 = \mathbf{n}^T\mathbf{x}_p$, and $\rho = \mathbf{n}^T\mathbf{x}_p/\mathbf{n}^T\mathbf{u}_0$. The intersection has homogeneous coordinates $(\rho\mathbf{u}_0, 1)^T = (\mathbf{n}^T\mathbf{x}_p\mathbf{u}_0, \mathbf{n}^T\mathbf{u}_0)^T$. Projecting this into the second image gives:

$$\mathbf{u}_1 = \mathtt{R}[\mathtt{I}|-\mathbf{t}]\begin{pmatrix}\mathbf{n}^T\mathbf{x}_p\mathbf{u}_0\\ \mathbf{n}^T\mathbf{u}_0\end{pmatrix} = \mathtt{R}[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T]\mathbf{u}_0 \tag{2}$$

where $\mathtt{I}$ is a 3x3 identity matrix. Therefore idealised image positions in the two camera images are related by the homography:

$$\mathtt{H}_I = \mathtt{R}[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T]. \tag{3}$$

If the cameras have intrinsic matrices $\mathtt{C}_0$ and $\mathtt{C}_1$, the homography in pixel coordinates is:

$$\mathtt{H} = \mathtt{C}_1\mathtt{R}[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T]\mathtt{C}_0^{-1}. \tag{4}$$

This equation can be used to predict the transformation of a feature's visual appearance between its template image and the current camera image. Finding the difference between this prediction and the actual appearance of the feature, and the associated correction of the surface normal estimate is done using the image registration method described in the next section. This method provides an *inverse compositional* adjustment; i.e. the original image transformation, $\mathtt{H}_{n-1}$, is updated with an adjustment $\mathtt{H}_i$, to give a new estimate of the transformation $\mathtt{H}_n$ such that:

$$\mathtt{H}_n = \mathtt{H}_{n-1}\mathtt{H}_i^{-1}\ . \tag{5}$$

We want to parameterise $\mathtt{H}_i$ in terms of changes to the surface normal $\mathbf{n}$. The normal is adjusted by adding a vector $\mathbf{d}$ to it, where $\mathbf{d} = \alpha\bar{\mathbf{x}}_p^1 + \beta\bar{\mathbf{x}}_p^2$. The 3-vectors $\bar{\mathbf{x}}_p^1$ and $\bar{\mathbf{x}}_p^2$ can be chosen as any unit vectors which are perpendicular to each other and to $\mathbf{x}_p$, and α and β parameterise the amount of adjustment. The additive adjustment $\mathbf{d}$ is sufficient to vary the surface normal over the range of orientations for which the plane would be visible in Camera 0. Using Equations 4 and 5 the adjustment transform $\mathtt{H}_i$ becomes:

$$\begin{aligned}
\mathtt{H}_i &= \mathtt{H}_n^{-1}\mathtt{H}_{n-1} \\
&= \left[\mathtt{C}_1\mathtt{R}[(\mathbf{n}+\mathbf{d})^T\mathbf{x}_p\mathtt{I} - \mathbf{t}(\mathbf{n}+\mathbf{d})^T]\mathtt{C}_0^{-1}\right]^{-1}\left[\mathtt{C}_1\mathtt{R}[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T]\mathtt{C}_0^{-1}\right] \\
&= \mathtt{C}_0[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T + \mathbf{t}\mathbf{d}^T]^{-1}[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T]\mathtt{C}_0^{-1} \\
&= \mathtt{C}_0[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T + \mathbf{t}(\alpha\bar{\mathbf{x}}_p^1 + \beta\bar{\mathbf{x}}_p^2)^T]^{-1}[\mathbf{n}^T\mathbf{x}_p\mathtt{I} - \mathbf{t}\mathbf{n}^T]\mathtt{C}_0^{-1}. \qquad (6)
\end{aligned}$$

2.2 Gradient-based Image Registration

Inverse compositional gradient-based image registration methods aim to refine an estimate of the transformation between two images I_T and I_n by minimising the difference between I_n warped by the inverse of the current warp estimate and I_T warped by an incremental warp $W_i(\mathbf{x};\mathbf{p})$, through variation of the parameter vector $\mathbf{p}$. The aim of alignment is to find the parameter vector $\mathbf{p}$ for which the following is close to true for all pixels in a region:

$$I_n(W_{n-1}(\mathbf{x})) = I_T(W_i(\mathbf{x};\mathbf{p})) \approx I_T(\mathbf{x}) + \nabla I_T \frac{\partial W_i}{\partial \mathbf{p}}\Delta\mathbf{p}\ . \qquad (7)$$

The estimation of $\mathbf{p}$ can be done probabilistically by treating the difference between the estimate $I_n(W_{n-1}(\mathbf{x}))$ and the actual template image $I_T(\mathbf{x})$ as an uncertain measurement which tells us something about the state $\mathbf{p}$, and applying Bayes rule to calculate a posterior estimate of $\mathbf{p}$. We use the measurement model:

$$z = I_n(W_{n-1}(\mathbf{x})) - I_T(\mathbf{x}) = \nabla I_T \frac{\partial W_i}{\partial \mathbf{p}}\Delta\mathbf{p} + n, \qquad (8)$$

where the n term represents zero mean measurement noise. Calculation of a posterior estimate of $\mathbf{p}$ is discussed in detail in our previous work [6, 7]. If the prior estimate of $\mathbf{p}$ is $\mathbf{p}_{n-1}$ with covariance $\Lambda_{\mathbf{p},n-1}$, and the measurement noise n has variance σ_z^2, the posterior distribution of $\mathbf{p}$ has mean $\mathbf{p}_n$ and covariance $\Lambda_{\mathbf{p},n}$ given by:

$$\mathbf{p}_n = \Lambda_{\mathbf{p},n}\left[\Lambda_{\mathbf{p},n-1}^{-1}\mathbf{p}_{n-1} + \sum_i \sigma_z^{-2}\left(\nabla I_T \frac{\partial W_i}{\partial \mathbf{p}}\right)^T (I_n(W_{n-1}(\mathbf{x}) - I_T(\mathbf{x}))\right] \qquad (9)$$

$$\Lambda_{\mathbf{p},n} = \left[\Lambda_{\mathbf{p},n-1}^{-1} + \sum_i \sigma_z^{-2}\left(\nabla I_T \frac{\partial W_i}{\partial \mathbf{p}}\right)^T \left(\nabla I_T \frac{\partial W_i}{\partial \mathbf{p}}\right)\right]^{-1}, \qquad (10)$$

where the summations are over a set of pixels of interest. This is an information-weighted sum of the prior and measurement. In the case of correcting an estimate of surface normal, the parameter vector $\mathbf{p}$ is the vector $(\alpha,\beta)^T$ which has a prior $\mathbf{p}_{n-1}$ of $(0,0)$ with a covariance $\Lambda_{\mathbf{p},n-1}$ which is the projection of the uncertainty in the normal estimate onto the $\bar{\mathbf{x}}_p^1$-$\bar{\mathbf{x}}_p^2$ plane. In this work we set the variance of n (i.e. σ_z^{-2}) to a constant value as

we have previously found that more sophisticated noise models do not give a noticeably improved result [6]. In practice the value of $\mathbf{p}$ is calculated by iteratively applying Equations 5 and 9, and maintaining the prior $\mathbf{p}_{n-1}$ to represent the value at the start of iteration, while keeping the covariance $\Lambda_{\mathbf{p},n}$ constant.

For this application, where we need to balance the variability in the warp parameters (see below) and are trying to maintain an estimate of the surface normal with first order uncertainty, use of a probabilistic alignment method, which does not add significantly to the processing cost [6, 7], is essential for a stable and useful result. We use an inverse compositional method for its efficiency [1], although the fact that W_i changes in each frame means that some of its efficiency advantage over other methods is lost.

The warps W_i and W_{n-1} are a function of the relative position of the cameras and feature plane in the scene, defined by the set of parameters Φ $(\mathbf{n}, \mathbf{x}_p, \bar{\mathbf{x}}^1_p, \bar{\mathbf{x}}^2_p, \mathtt{R}, \mathbf{t}, \mathtt{C}_0, \mathtt{C}_1)$. The calculation described above assumes that $\mathbf{n}$ is the only uncertain value among these, which in practice is not the case. As a consequence, the projection of $\mathbf{x}_p$ into the template view does not in general correspond to the feature's detected position in that view, and to a lesser extent, the projection into the current image does not correspond to the feature's position there obtained by template matching. The two parameter warp model used does not have enough freedom to accommodate the effect of these additional errors in Φ.

This is dealt with by artificially altering the principal point values in camera calibration matrices $\mathtt{C}_0$ and $\mathtt{C}_1$ to make these projections coincide, and by extending the aligning warp with a 2D translation (τ_x, τ_y). This is equivalent to changing the incremental warp $\mathtt{H}_i$ to $\mathtt{H}_T\mathtt{H}_i$, where $\mathtt{H}_T$ represents a translation (τ_x, τ_y) — allowing the template image to translate before other transformations are applied. The translation parameters are only allowed to vary over a small range by setting the prior covariance to 1 pixel squared. This implicitly assumes that besides image translation the effect of changing Φ is less significant on patch appearance than the effect of changing $\mathbf{p}$, which was the case in our results.

2.3 Resampling Issues

Equation 9 implies the resampling of the alignment image I_n back to the template image I_T, which we do using bilinear interpolation. The result of this is that an unprocessed image, I_T, is compared to an image which has been resampled, $I_n(W_{n-1}(\mathbf{x}))$, and will be more blurred as a result. By resampling an image under a small homography, then resampling again under the inverse homography, and comparing to a blurred version of the original image, we determined that under a range of small homographies the effect of bilinear interpolation is equivalent to Gaussian smoothing with a standard deviation of about 0.5 pixels. This smoothing is applied one-off to the template image patch I_T.

As the relative displacement of the cameras becomes large another resampling issue arises. If the template image patch corresponds to a frontoparallel sampling of the surface but the current image is a view of the surface from a shallow angle, or a view from a lot further away, the current image will be missing some of the high frequency image details present in the template image. This can again be remedied by smoothing the template image, but since in this case the smoothing is a function of the homography $\mathtt{H}_{n-1}$ which is different for every alignment, and is in general non-isotropic, its implementation is relatively costly. For real-time operation the option exists of creating a representative set of 2D Gaussian blurred template image patches I_T, and using the one whose smoothing effect is most similar to the current value of $\mathtt{H}_{n-1}$. This can again be estimated empirically.

3 Maintaining the Surface Normal Estimate

In this section we explain how the difference found by gradient-based image registration between a pre-warped or "predicted" template and the actual current image state can be used to update the estimate of surface normal for a feature.

3.1 Filtering Surface Normal Estimates

Initially, the surface normal of each point is assigned to a "best guess" orientation facing the camera ($\mathbf{n}$ is set to the direction from $\mathbf{x}_p$ to the camera, and α and β are set to zero). To obtain an initial uncertainty for this assignment we simulated the effect of projecting a large number of randomly oriented planar patches into a camera. The resulting distribution of the projected image area as a function of α and β approximates the probability that a pixel chosen at random in an image comes from a scene surface of a particular orientation. The distribution is a bell shaped curve with a mean of zero, but has more density in the tails than a Gaussian. Because of this we set the initial uncertainty in α and β to quite a high initial value. The initial uncertainty in the translation part of the warp (τ_x, τ_y) is set to one pixel, and we set the standard deviation of the measurement noise (σ_z) to seven grey levels, which is representative of the error after a successful image alignment.

For each surface normal an independent state estimate and covariance is maintained. To update a normal estimate, the state and covariance are first transformed into the coordinate frame defined by the basis $(\mathbf{x}_p, \bar{\mathbf{x}}_p^1, \bar{\mathbf{x}}_p^2)$ for the alignment calculation, and then after the alignment are transformed back to the global coordinate frame. The covariance of the normal estimate is actually only ever non-zero in the plane perpendicular to the line joining the template image camera to the feature's position estimate.

Since the position of the camera from which a feature's template was first initialised is a critical part of the surface normal calculation, this is added to the SLAM state vector so that it can be updated if necessarily through correlations with the scene structure. This should not increase the processing cost of the filter significantly in the long term because postponement [5] can be used to limit the resulting update to times when the orientation is to be re-estimated, and because once we have an acceptably accurate estimate of the surface normal the initial camera position can be removed from the filter.

4 Results

The system described runs successfully in real-time (30Hz) on images from a firewire camera at 320×240 resolution and with all processing carried out on a 3GHz Pentium processor under Linux. We illustrate typical results with two sequences in this section.

4.1 Path Scene

The image on the left of Figure 2 shows part of a sequence captured with a hand-held camera viewing flat ground paved with concrete bricks. The image on the right represents the reconstruction in locally planar patches after around 600 frames of real-time tracking. The camera viewed the ground from an angle of about 45° while moving along an approximately circular arc through 90°. The wire-frame quadrilaterals in the figure show the initialised surface orientation of each patch and the textured patches show the final

result. The patch orientation estimates typically reach correct-looking values after performing the alignment two or three times. The yellow ellipsoids visible at some features represent uncertainty in feature position from the SLAM state vector (these are 3 standard deviation confidence regions). Some features with large ellipsoids have been measured only a small number of times and still have uncertain positions; the surface orientation estimates have similarly not yet settled down to accurate values. Features like **A** and **B** however have achieved good normal estimates while still having quite uncertain positions.

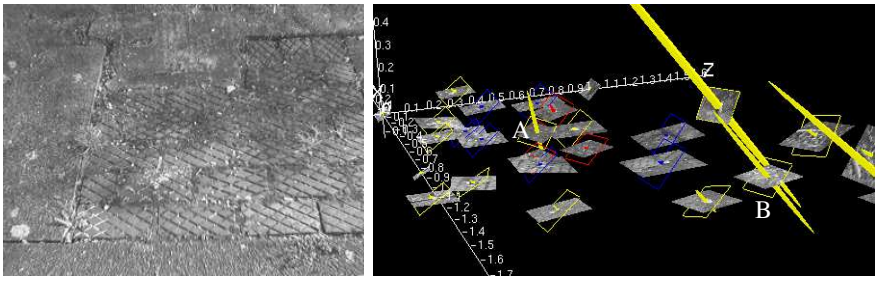

Figure 2: An image and structure from an outdoor sequence

At 30Hz, 33ms (including image acquisition and graphical rendering in our system) is available to process each image. In this experiment, processing took about 25ms per frame initially, rising to 30-35ms when the map had built up to 30 features. This rise is partly because the SLAM filter is wastefully maintaining the original camera position from which each feature was initialised (seven parameters), and we do not at present remove these from the state as described in Section 3.1. The alignment calculation takes 2ms per feature, running on the moderate gradient pixels in a 25 pixel square image area. Computational cost is highly dependent on the number of pixels used and the method may still be reliable with fewer pixels. Because the alignment calculation is the last operation each frame it could be done selectively after checking the processing time available.

To test the benefit to the SLAM system of estimating the surface normal, compared with the simpler alternative of sticking always with the initial normal direction parallel to the original viewing direction of each feature, we recorded the number of times correlation matching fails in each case. For this sequence the failure rate drops from 30.5% to 15.9% when surface orientations are updated, indicating as expected that with significant camera motion the appearance of features is much better predicted with good normal estimates.

4.2 Desktop Scene

Figure 3 shows images and structure from a second sequence, featuring a scene which includes several real planes at different orientations and regions which are less locally planar. Results with features which would not appear to fit the locally planar assumption varied; the crossed wires (labelled **A**), although non-coplanar, benefited from the extra freedom granted by the image alignment incremental warp, which is essentially a non-isotropic scaling, and were tracked well. Other features, such as the pair of features labelled **B** which lie on a severe depth discontinuity, do not benefit. In this case the surface normal estimate changes rapidly, causing tracking to become unstable.

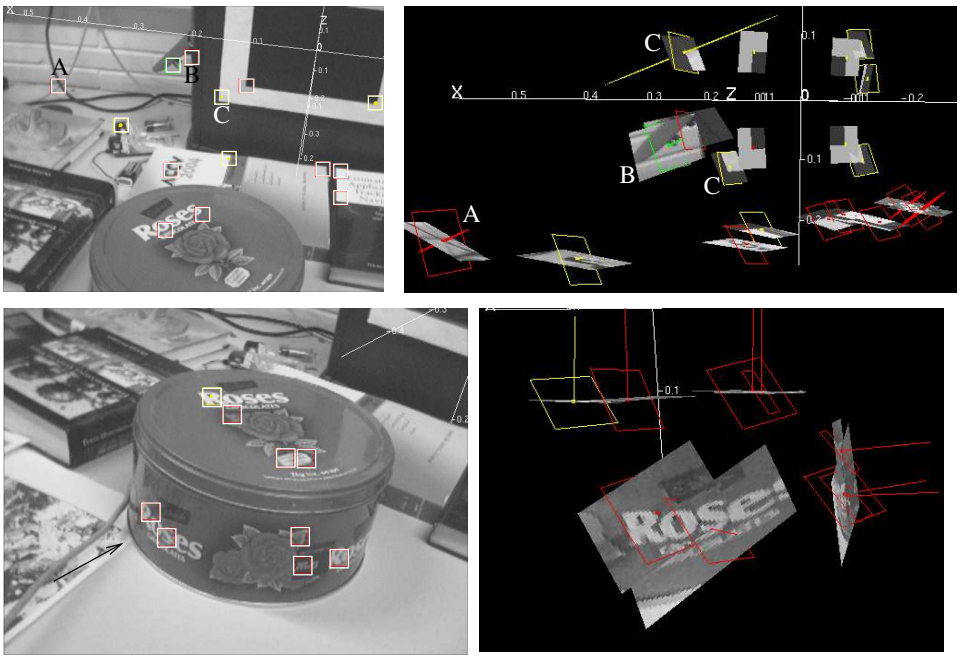

Figure 3: Images and reconstructions from the desktop sequence

Another failure mode arises when a feature becomes occluded. As the occlusion begins, the feature's appearance changes on one side, which causes the surface normal estimate to change strongly to compensate, making it harder to rematch the feature if it later becomes visible again. A future improvement would be to detect automatically during the alignment when such occasions arise and elect not to update the normal estimate.

The appearance of some features (such as the pure black-on-white corners labelled **C** in the figure) is invariant to many transformations and therefore cannot provide the information needed to estimate surface orientation. Because of the prior used in image alignment the surface normal estimate for these features does not become unstable, but the unestimated normal can be misleading in trying to interpret scene structure.

The bottom left and right images of Figure 3 show the successful reconstruction of several features on a sweet tin. The image on the right is from a viewpoint indicated by the arrow in the left image: where the two features on the left side of the tin are directly facing the camera, the three features on the top appear horizontal, and the three features on the right appear vertical. The line extending from each patch indicates the surface normal direction. Features on other objects in the scene have been removed for clarity.

5 Conclusion

We have demonstrated the benefits of a 3D locally planar representation in real-time point-based structure from motion, and of estimating surface orientation over time. This allows the appearance of features to be predicted much more accurately during camera movement, which makes it possible to track the features reliably for longer, resulting in a more

stable result. A richer structural map containing surface orientation estimates, as well as point positions, it also helps interpreting the scene. Indeed a similar method to that described here might equally be beneficial in an offline structure from motion system.

The method described attempts to estimate and account for the projective variance of planar features. The alternative is to match features using an invariant descriptor, and several have recently be proposed in the literature. In situations where the appearance of features has changed unpredictably (when the position of a camera is completely unknown), such a method is very powerful. However, when as in our system we have running estimates of camera and feature locations from SLAM it is arguably more informative and certainly more efficient to model these distortions explicitly whenever possible.

A future issue is to study the effect of the size of planar patches used, since surely the scale of these features should be set dynamically depending on factors including the 3D smoothness of the scene and the scales at which salient textures appear.

Acknowledgements: This work was supported by EPSRC Grant GR/R89080/01 and an EPSRC Advanced Research Fellowship to AJD.

References

[1] S. Baker and Iain Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 1. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[2] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the Ninth International Conference on Computer Vision ICCV'03*, pages 1403–1410, Nice, France, 2003.

[3] H. Hattori and A. Maki. Stereo matching with direct surface orientation recovery. In *Ninth British Machine Vision Conference*, pages 356–366, Southampton, UK, September 1998.

[4] H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 192:1–18, 2003.

[5] J. Knight, A. Davison, and I. Reid. Towards constant time slam using postponement. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 406–412, Maui, 2001.

[6] N. D. Molton, A. Davison, and I. Reid. Parameterisation and probability in image alignment. Report OUEL 2266/03, Department of Engineering Science, University of Oxford (Available from http:/www.robots.ox.ac.uk/˜ndm/), August 2003.

[7] N. D. Molton, A. Davison, and I. Reid. Parameterisation and probability in image alignment. In *Proceedings of the Sixth Asian Conference on Computer Vision*, pages 192–197, Jeju, South Korea, January 2004.

[8] D Nister. Preemptive ransac for live structure and motion estimation. In *Proceedings of the Ninth International Conference on Computer Vision*, pages 199–206, Nice, France, 2003. IEEE Computer Society Press.

[9] S. Se, D. Lowe, and J. Little. Mobile robot localisation and mapping with uncertainty using scale-invariant visual landmarks. *The International Journal of Robotics Research*, 21(8):735–757, 2002.