

Real-Time Visual SLAM with Resilience to Erratic Motion

Mark Pupilli and Andrew Calway
Department of Computer Science
University of Bristol, UK
{pupilli, andrew}@cs.bris.ac.uk

Abstract

Simultaneous localisation and mapping using a single camera becomes difficult when erratic motions violate predictive motion models. This problem needs to be addressed when visual SLAM algorithms are transferred from robots or mobile vehicles onto hand-held or wearable devices. In this paper we describe a novel SLAM extension to a camera localisation algorithm based on particle filtering which provides resilience to erratic motion. The mapping component is based on auxiliary unscented Kalman filters coupled to the main particle filter via measurement covariances. This coupling allows the system to survive unpredictable motions such as camera shake, and enables a return to full SLAM operation once normal motion resumes. We present results demonstrating the effectiveness of the approach when operating within a desktop environment.

1. Introduction

The problem of estimating location using vision comes in a variety of guises. Recently there has been a trend towards developing real-time ‘pick up and play’ single camera systems, in which the principles of simultaneous localisation and mapping (SLAM) are used. This gives the potential for highly portable camera tracking over wide areas for applications such as augmented reality and location aware devices in wearable computing. It is a challenging area of research, due the inherent ambiguities in visual measurements and the fact that for real-time operation they need to be processed immediately to update estimates of camera pose and scene structure. This rules out using batch optimisation and instead requires sequential processing such as that provided by recursive Bayesian filters.

Previous approaches have been based on variants of the Kalman filter [1, 2, 3]. A key element of these is the use of predictive motion models such as constant velocity. When such models are not violated the filters are extremely efficient since the predictive power of the model minimises

search areas for costly image processing operations and allows outlier rejection. However, when the models are violated, the filters can become unstable and diverge. Such violations can often occur when cameras are hand-held or wearable simply due to the nature of human movement. For example, users wearing head mounted displays for augmented reality often want to make rapid and unexpected head movements and this has been reported to cause problems [5].

The mechanisms by which erratic motion can cause recursive estimators to fail are complex. In the worst case, use of highly predictive motion models will lead to rapid divergence from the true state of affairs. In the best case, no suitable measurement will be found in the predicted search region. This is followed by uncertainty expansion, which in principle should lead to the correct feature being located and convergence of the filter. Unfortunately this is unlikely to be the case. In reality the enlarged search regions lead to multiple measurement hypotheses. Uni-modal estimators such as the Kalman filter are then forced to pick a measurement and selecting incorrectly leads to divergence in localisation and mapping and hence system failure. If the correct measurement can be found the system may be so far from the last good estimate that linearisation errors are large or convergence to the correct state is slow.

In this paper we describe a novel approach to visual SLAM which aims to address these problems. Specifically, our goal is a filter which is resilient to erratic motions, in the sense of being able to resume full SLAM operation following such motions and avoiding system failure. Motivated by the observations in the previous paragraph, we adopt two key principles. First, that (at least short-term) multiple hypotheses of camera pose need to be retained; and second, that structure mapping needs to be postponed when the camera undergoes erratic motion in order to avoid divergence. To achieve this, we have utilised the particle filtering framework developed by Pupilli and Calway [7] for real-time camera tracking. The particle filter retains multiple hypotheses for the camera pose and hence provides resilience to erratic motion. To this we have added a mapping exten-

sion which allows simultaneous estimation of feature depth and camera pose. It is based on auxiliary unscented Kalman filters (UKF), which update the depths in tandem with the main particle filter. The UKF also allows for efficient use of the depth estimates and their uncertainty for localising the camera and hence simultaneous real-time operation. Crucially, the measurement covariances for the auxiliary filters are derived from the sample set within the particle filter, enabling the uncertainty in camera pose to be used directly in the map updates. It is this coupling between the filters that gives the potential for resilience to erratic motion when undertaking full SLAM.

The approach can be compared with the FastSLAM algorithm developed by Montemerlo *et al.* for 2-D robot navigation [6]. They also use Kalman estimators for mapping, although attach a bank of estimators to each particle to allow independent mapping of landmark features. Thus each particle has its own local map in contrast to the single map maintained by our auxiliary UKFs. FastSLAM has the advantage of scalability over previous Kalman filter approaches, although the lack of a direct link between the uncertainty in localisation and the updating of the map (each particle represents a single trajectory) may lead to divergence during unpredictable movements. This may be overcome by the use of large particle sets, although this is likely to prohibit real-time operation given the size of the particle state. This is addressed by our maintenance of a single map via the auxiliary UKFs, although, as discussed later, this has the drawback that full covariance information across the map is not maintained, hence limiting the algorithm’s ability to perform wide area SLAM.

In the next section we outline the particle filtering framework for camera localisation. Full details can be found in [7]. Our SLAM extension is then described in Section 3 and we present results illustrating the performance of the algorithm in Section 4.

2. Camera Localisation Using a Particle Filter

For camera localisation, we are interested in recursively estimating the pose (3-D position and orientation) of the camera for each image frame k . We use a state space model of the form $\mathbf{x}_k = \{\mathbf{t}_k, \mathbf{q}_k\}$, consisting of a translation vector and a quaternion (rotation) to parameterise the camera’s state w.r.t. the world co-ordinate system. For now, we also assume that we have a set of 3-D scene points $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_M\}$ whose locations are known in the world co-ordinate frame. For a particular camera \mathbf{x}_k we can compute the projection of every scene point using the following equation:

$$\mathbf{u}(\mathbf{z}_m, \mathbf{x}_k) = \Pi(R_k \mathbf{z}_m + \mathbf{t}_k) \quad (1)$$

where R_k is the rotation matrix derived from the quaternion \mathbf{q}_k and Π is the standard pin-hole projection for a calibrated

camera.

At each frame we take measurements \mathbf{y}_k from the image and we denote the set of such observations up to the current frame $\mathbf{y}_{1:k}$. The particle filter then provides recursive approximations to the posterior density $p(\mathbf{x}_k | \mathbf{y}_{1:k}, Z)$ as a weighted sample set $\{(\mathbf{x}_k^1, w_k^1), \dots, (\mathbf{x}_k^N, w_k^N)\}$. The weights are proportional to the likelihood $p(\mathbf{y}_k | \mathbf{x}_k, Z)$ and the sum of the weights is $\sum_{n=1}^N w_k^n = 1$. At each step of the filter particles are re-sampled from the transition density $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ which defines the motion model. In order to deal with erratic motions this was chosen in [7] to be the maximum distance model commonly used with legged robots [4].

The likelihood $p(\mathbf{y}_k | \mathbf{x}_k, Z)$ is based on an inlier count for each particle determining how many scene points \mathbf{z}_m project close to a relevant measurement. Specifically, each scene point \mathbf{z}_m has an associated reference template and this is used to generate a correlation field in the current frame. Measurement candidates \mathbf{y}_{km} which have high correlation values are then selected. The likelihood is then given by the following function

$$p(\mathbf{y}_k | \mathbf{x}_k, Z) \propto \exp\left(-\sum_{i=1}^M \prod_{\mathbf{u} \in \mathbf{y}_{km}} d(\mathbf{u}, \mathbf{z}_m, \mathbf{x}_k)\right) \quad (2)$$

where $d(\mathbf{u}, \mathbf{z}_m, \mathbf{x}_k)$ indicates whether the point \mathbf{z}_m is an inlier or outlier with respect to the observation at \mathbf{u} and the camera pose \mathbf{x}_k

$$d(\mathbf{u}, \mathbf{z}_m, \mathbf{x}_k) = \begin{cases} 1 & \text{if } \|\mathbf{u} - \mathbf{u}(\mathbf{z}_m, \mathbf{x}_k)\| > \epsilon_d \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The threshold ϵ_d represents the uncertainty in the correlation values and in the location of the 3-D scene point \mathbf{z}_m . In the SLAM extension described below, the latter is obtained from the covariances carried by the auxiliary UKFs (see Section 3.4). Particle annealing was also used in [7] to refine the posterior density and also to re-weight the multiple candidates so that the most likely measurement from the set \mathbf{y}_{km} can be associated with \mathbf{z}_m in the presence of clutter. Finally, an estimate of the camera pose $\bar{\mathbf{x}}_k$ can then be obtained from the mean of the annealed distribution.

3. SLAM Extension

The above localisation framework relies on knowledge of the 3-D scene points Z . In practice these need to be obtained simultaneously with the camera localisation. We assume that we can initialise the localisation using a known 3-D test pattern [3, 7]. The task then is to dynamically select new features to map and to recursively estimate their corresponding depths. Crucially, the estimates and their uncertainty need to be utilised in the localisation; only by doing this can camera tracking take place over wide areas.

3.1. Uncertainty in Feature Mapping

For a selected feature, we estimate its depth using an auxiliary unscented Kalman filter. Ideally, one may consider using a full particle filter implementation for estimating both pose and structure, as in [8] for example, although the computational demands prohibit real-time operation. In contrast, use of the UKF allows for fast updating of the map. Moreover, for our purposes, it is sufficient to maintain a mean and covariance representation for the mapping providing that we can incorporate the uncertainty in camera localisation directly within the UKF update and effectively postpone mapping when localisation becomes uncertain.

We parameterise the location of a feature in the world co-ordinate system in terms of direction and depth $d_r \mathbf{v}_r$, where \mathbf{v}_r is the unit vector in the direction of the feature. For each new feature, \mathbf{v}_r is known (this amounts to knowing the direction from the camera centre of projection to the feature's location in the image plane). However, we are uncertain of the depth of the feature d_r so this must go in our state space model. We are also uncertain about the camera pose so this also needs to be included. If multiple features are introduced in the same frame we can extend the state by appending their associated depths, giving the state model $\mathbf{c}_r = \{\mathbf{t}_r, \mathbf{q}_r, d_r^1, \dots, d_r^M\}$, where d_r^m is the depth of the m th feature introduced in frame r . From this state we can compute the 3-D location of a feature in the world co-ordinate frame using the following equation:

$$\mathbf{z}_m(\mathbf{c}_r) = d_r^m R_r \mathbf{v}_r + \mathbf{t}_r \quad (4)$$

Combining multiple features in the state enforces the obvious constraint that features introduced in the same reference frame share a common camera pose (and hence share the uncertainty in this pose).

It might seem that our state space is over parameterised since each feature only has 3 degrees of freedom in its Cartesian uncertainty. However, if we consider the case when four features are introduced per frame then we actually save a dimension over a Cartesian representation (3×4 vs. $7+4$), a saving which increases linearly by adding further features. Our representation also parameterises the state along axes which are physically relevant which in turn leads to covariances that are more diagonal than would be the case with a Cartesian state. This is similar to a depth-bias representation of structure, the benefits of which are discussed in [1].

Once we have identified features in some reference frame we need to initialise the UKF mean and covariance for this feature set. The camera mean $\{\bar{\mathbf{t}}_r, \bar{\mathbf{q}}_r\}$ and covariance Σ_{c_r} for the reference frame is computed from the PF sample set $\{(\mathbf{x}_r^1, w_r^1), \dots, (\mathbf{x}_r^N, w_r^N)\}$. However, at this point we have no idea of the depth of the features so we cannot immediately initialise means and variances. Instead,

we store an initial 3-D reference ray for each feature, beginning at the camera's centre of projection, and extending through the pixel at which the template was located. In subsequent frames, we detect multiple candidate locations for this feature. Given the distribution over camera states in each subsequent frame k , represented as the sample set $\{(\mathbf{x}_k^1, w_k^1), \dots, (\mathbf{x}_k^N, w_k^N)\}$, we can define a number of rays for each hypothesised camera \mathbf{x}_k^n : one for each candidate template match. Each of these rays can be intersected with the reference ray. By assigning a weighted particle to each intersection along the reference ray a 1-D depth distribution is constructed. This distribution is recursively updated over a number of frames. Once this distribution becomes approximately Gaussian we can take the mean \bar{d}_r^m and variance $\sigma_{d_r^m}^2$ and hand estimation of the feature location over to the UKF. This can be compared to the factored sampling which was used to initialise depth in [3], although it has the key advantage of avoiding a fixed depth prior.

In practice each feature is introduced into the UKF state at different times. For clarity we will ignore that the state is of variable size, in which case the initial UKF mean and covariance is then as follows:

$$\begin{aligned} \mathbf{c}_0 &= [\bar{\mathbf{t}}_r, \bar{\mathbf{q}}_r, \bar{d}_r^1, \dots, \bar{d}_r^M] \\ \Sigma_0 &= \begin{bmatrix} \Sigma_{c_r} & 0 \\ 0 & \Sigma_{d_r} \end{bmatrix} \\ \Sigma_{d_r} &= \text{diag}[\sigma_{d_r^1}^2, \dots, \sigma_{d_r^M}^2] \end{aligned}$$

Note that the off-diagonal blocks of Σ_0 are zero and the bottom right block corresponding to the depth variances is diagonal. Hence, the depth estimates are not initially correlated with the camera state or each other. In contrast, Σ_{c_r} is generally a full covariance matrix.

3.2. UKF Measurement Model

The state evolution model for each UKF is simply the identity with no noise component since we are estimating constants (the scene structure is assumed to be rigid). Now let us consider how to formulate the measurement model. At each frame k we are going to use the UKF to project our uncertainty into the mean camera estimate from our particle filter $\bar{\mathbf{x}}_k$. This constitutes the basis for our UKF measurement model and is given formally by:

$$\mathbf{h}(\mathbf{c}_r) = \begin{bmatrix} \mathbf{u}(\mathbf{z}_1(\mathbf{c}_r), \bar{\mathbf{x}}_k) \\ \vdots \\ \mathbf{u}(\mathbf{z}_M(\mathbf{c}_r), \bar{\mathbf{x}}_k) \end{bmatrix} + \mathbf{n}_k \quad (5)$$

where the functions \mathbf{u} and \mathbf{z}_m are defined in equations 1 and 4, respectively. We do not need to worry about the Jacobian for \mathbf{h} since the UKF uses unscented transforms to propagate the covariance through the non-linear measurement function [9].

The (time varying) measurement noise \mathbf{n}_k comes from two sources. The first source is simply the pixel based error in template matching and is not time varying. We denote its covariance R_{pixel} and assume it to be isotropic and Gaussian. The second is due to the uncertainty of our current camera state estimate $\bar{\mathbf{x}}_k$. We estimate this measurement noise covariance for every frame k by projecting each feature’s estimated position $\mathbf{z}_m(\mathbf{c}_r)$ into every camera particle in our sample set. The weighted sample covariance of the resulting projections can be computed. The sum of these two sources leads to the following measurement noise covariance:

$$R_k = R_{pixel} + Cov[\mathbf{u}(\mathbf{z}_m(\mathbf{c}_r), \mathbf{x}_k^n)] \quad (6)$$

where $Cov[\]$ is computed over $1 \leq m \leq M$ and $1 \leq n \leq N$. Note that this is a full covariance matrix with dimension $2M \times 2M$ where M is the number of features in the reference frame. The computation of this noise component makes the UKF robust in the presence of erratic motion. Intuitively, the degree to which the filter believes measurements is determined by the ‘spread’ of the particle distribution. When motion is erratic the particle filter diverges temporarily and this in turn postpones the updating of the mean and covariance in the UKF. When the camera stabilises again the particle filter converges and the auxiliary filters can make more reliable updates.

3.3. Clutter and Occlusions

During erratic motions guided search regions can become large which makes it likely that multiple candidate feature matches can be found. This problem of multiple measurement hypothesis (clutter) is easily dealt with in this framework by allowing particle annealing [7] to perform data association: the measurement used by the UKFs is this best re-weighted measurement. If no measurements are found the UKF is not updated so that occlusions can be dealt with. In the case when measurements can only be found for some of an auxiliary filter’s features, partial corrections are made to the UKF mean and covariance by ignoring the appropriate elements of the measurement covariance and cross correlation matrix in the UKF prediction. This has the desirable effect of allowing a feature estimate to become more certain without actually observing it by reducing the camera pose uncertainty in the feature’s reference frame but leaving the depth uncertainty untouched.

3.4. Tracking with New Features

Features which fail to converge quickly enough are discarded. Any features which pass this probationary period are then allowed to contribute to the PF camera location estimates. The mean feature locations are projected into each camera sample and inlier counting is performed as described earlier. The inlier radius used for the inlier count

is determined to be one standard deviation of the predicted measurement covariance. The reason for this apparently arbitrary choice is related to the information we expect to supply to the particle distribution during the weight assignment process. Clearly if we make the inlier region too large for a particular feature then all particles will get the same contribution from this feature: no information is gained. If we make it too small all particles will get zero weight: no information is gained. Now, the UKF predicted measurement covariance is, at its minimum (when the feature location is certain), equal to the PF measurement covariance plus a small amount of pixel noise (equation 6 and see Figure 2). By setting the inlier region to a 1σ gating we can be assured that the feature will provide some information to the particle distribution *provided the UKF predicted measurement covariance has converged close enough to the lower bound R_k* . We can in fact go further and say that unless the measurement covariance has converged close to R_k then this feature is not providing any information to the particle distribution so we should not waste computational resources by including it in the weighting process. This is still in some sense an empirical choice but it will suffice until an information theoretic correct solution can be developed.

4. Results

Experiments were carried out in a desktop environment using a calibrated hand-held web-cam with a resolution of 320×240 pixels. Tracking is initialised with four known points corresponding to the corners of a black rectangular calibration pattern (see figures). We used 500 camera samples in the particle filter. The algorithm operates in real-time with a frame rate 30 frames per second when 10 features are being mapped and tracked.

Figure 1 illustrates the operation of the algorithm as it successfully tracks the camera as it is moved arbitrarily over a desk scene. The left hand images show the 3-D localisation and mapping and the right hand images show the projection of the mean and covariances of the mapped scene points in the mean camera view (shown in yellow) for selected frames. The former show the mean camera trajectory and the particle weights for the location in the current frame, where white indicates high weight, and also the ellipses representing the mapping uncertainty within the UKFs. Note the convergence of the ellipses as new features are successfully mapped and that SLAM operation successfully continues away from the calibration pattern (row 2). Part way through this sequence the camera was shaken and localisation is temporarily disrupted as shown in row 3. Note the resultant spread of the particle cloud and, importantly, the stability of the mapped features, indicating that mapping has been successfully postponed during the shake. As the camera stabilises again, full SLAM operation continues. The dispersion of the particle cloud can also be seen in the

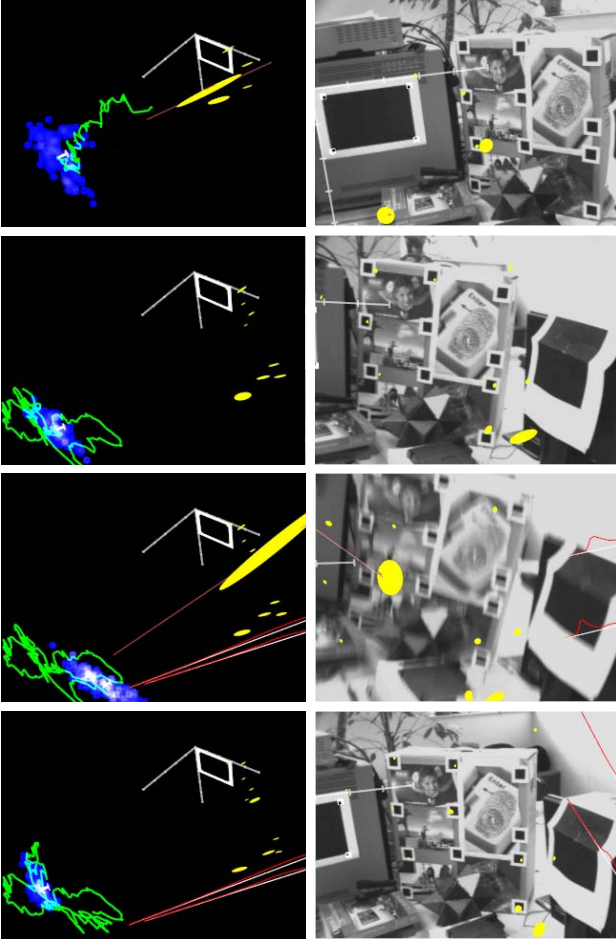


Figure 1. Results for selected frames during visual SLAM in a desktop environment: 3-D localisation and mapping (left) and projected mean and covariances of scene points into the mean camera view (right). Note the convergence of the mapping covariances and the successful recovery from camera shake (row 3).

left image of Figure 2 which shows the projections of the mean scene points into each camera particle. This can be compared with the tight distributions which result when normal motion resumes as shown in the right image. These images also show the measurement covariances within each UKF (blue and red) and also the covariances of the particle clouds (yellow). Note the convergence of the UKF covariances to that of the particle cloud. The red covariances indicate that feature mismatch has occurred which is prevalent during camera shake in the left-hand image. The spread of the camera particles can also be seen from the evolution of the estimated posterior each motion parameter obtained from the particle filter shown in Figure 3. Note the dispersion of particles around frame 700.

Figure 4 illustrates the ability of the algorithm to accurately map structure despite violent camera shake. In this case an experiment was set up with the initialisation rectan-

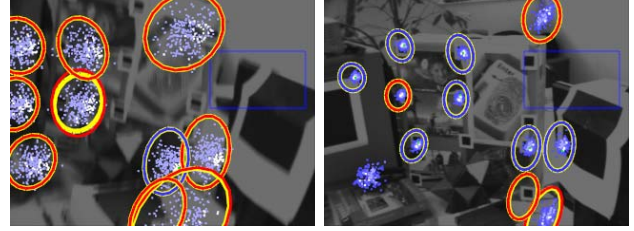


Figure 2. Projections of scene points into each camera particle and corresponding measurement covariances from the UKFs for two frames from the example in Figure 1.

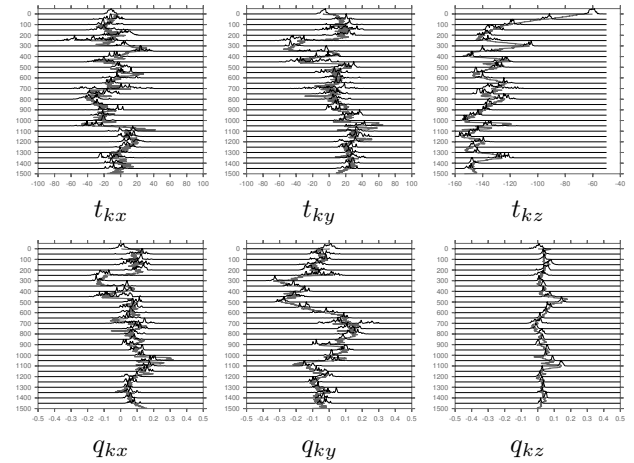


Figure 3. Estimated posteriors for the camera motion parameters: translation $\mathbf{t}_k = (t_{kx}, t_{ky}, t_{kz})$ and quaternion $\mathbf{q}_k = (q_{kx}, q_{ky}, q_{kz})$. Note the spreading of the distributions during camera shake around frame 700.

gle flat on a desk and a sugar sachet placed in its centre. A chess board comes into view after a few frames of tracking. The chess board is elevated with its top surface measured to be 7.7cm above the desk. In one frame we manually select 3 points on the chess board. In the very next frame we manually select two points on the chess board and one point on the sugar sachet. The intention here is to test how the different coupling affects the estimation of feature locations.

Row 1 in Figure 4 shows the newly introduced features (right) and their measurement covariances (left), and the colour coding shows the coupling. Tracking continues for a few frames and then the camera is shaken violently. There are a lack of good measurements in many frames because of motion blur. This leads to enlargement of the measurement covariance and correspondingly large search regions (row 2). Note that the chess board pattern here means there is much room for ambiguity in measurement of the new features because of the enlarged search regions. Once the shaking ceases all the features on the chess board are out of view but good measurements of the sugar sachet allow its depth to converge (row 3). Eventually, all the features come back

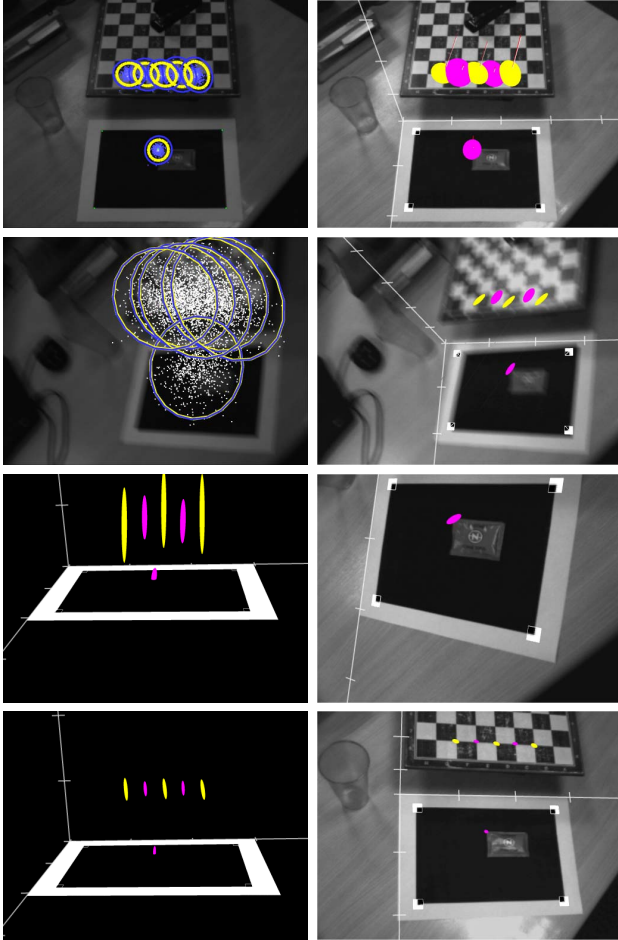


Figure 4. Results illustrating successful mapping despite severe camera shake. Note the dispersion of the camera particles at the onset of shake shown by the projections of mapped scene points in the left image of row 2.

into view (row 4) but the features which were coupled to the sugar sachet are more certain than the other three because they share their reference frame uncertainty with a tightly converged feature. The difference in convergence can be seen in the 3-D plots of the feature location covariance (row 3 and 4, left). The mean height estimates of all features are within 2cm of their measured elevation from the desk.

5. Conclusion

We have developed a visual SLAM algorithm which combines particle filtering and unscented Kalman filtering in a novel way. This combination was demonstrated to give the SLAM system resilience to erratic motions. The key component of the work is the coupling of the filters via the measurement covariances, which appears to provide an effective way of merging the flexibility of the particle filter with the computational efficiency of the UKF. There are

a number of areas for further research. At present, our formulation only utilises covariance amongst features initialised in the same frame. An obvious extension to this is to develop a representation based on full-covariance between the camera state and newly mapped features, using a single auxiliary UKF sitting on top of the particle filter. The latter would then act as an early warning system for violations of the motion model through its influence on the measurement covariance. Equally, violations of uni-modal assumptions can be detected by the particle filter which can perform the required data associations. Another important area of work relates to feature measurements. A significant limitation of our current system is its reliance on fixed appearance templates for matching features. Other visual SLAM algorithms have benefited significantly from the use of view point and scale invariant image features and we aim to incorporate similar mechanisms into our system in the future.

References

- [1] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.
- [2] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 24(4):523–535, 2002.
- [3] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. Int Conf on Computer Vision*, 2003.
- [4] J.-S. Gutmann and D. Fox. An experimental comparison of localization methods continued. In *Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems*, 2002.
- [5] G. Klein and T. Drummond. Tightly integrated sensor fusion for robust visual tracking. In *Proc. British Machine Vision Conference*, 2002.
- [6] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-SLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc Int Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- [7] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *Proc. British Machine Vision Conference*, 2005.
- [8] G. Qian and R. Chellappa. Structure from motion using sequential monte carlo methods. *Int Journal of Computer Vision*, 59:5–31, 2004.
- [9] E. Wan and R. van der Merwe. The unscented kalman filter for nonlinear estimation. In *Proc. IEEE Symp on Adaptive Systems for Signal Processing*, 2000.

Acknowledgement This work was funded by the UK EPSRC Equator IRC.