Particle Filtering for Robust Single Camera Localisation

Mark Pupilli and Andrew Calway

Department of Computer Science
University of Bristol, UK
{pupilli,andrew}@cs.bris.ac.uk

Abstract. This paper summarises recent work on vision based localisation of a moving camera using particle filtering. We are interested in real-time operation for applications in mobile and wearable computing, in which the camera is worn or held by a user. Specifically, we aim for localisation algorithms which are robust to the real-life motions associated with human activity and to the dynamic clutter encountered in real environments. Particle filtering provides greater generality than previous approaches, enabling it to deal with the multi-modal uncertainties characteristic of such operating conditions. We present an overview of the methodology and experimental results for different tracking scenarios, with and without prior knowledge of scene structure.

1 Introduction

The emergence of cheap, easy to use digital cameras and the increasing processing power available on small devices, has given new impetus to the development of algorithms for real-time localisation of a moving camera, i.e. frame by frame estimation of camera pose. If achieved reliably, then numerous potential applications open up, notably in areas of mobile and wearable computing, in which vision can play a central role, e.g. in augmented reality systems where rendered graphics need to be synchronised with incoming video streams. Although camera pose estimation has a long history in computer vision, and robust methods exist for off-line solutions, transferring these to real-time operation within real dynamic environments presents new research challenges.

Existing systems can be broadly classified into those utilising prior knowledge of 3-D structure, in the form of calibrated fiducial markers or 3-D scene models for example, and those which attempt simultaneous mapping of scene structure in tandem with tracking (SLAM). Most effort has been put into the former category, particularly in the area of augmented reality, where camera tracking in pre-calibrated environments using fiducial markers or salient features can be reasonably robust [6, 7]. Good tracking performance can also be obtained by using pre-defined 3-D models of objects in the scene, as in the systems developed by Drummond *et al* [5] and Vacchetti *et al* [9]. Simultaneous mapping of structure whilst tracking presents the biggest challenges and the systems are consequently less robust. Notable examples include that developed by Chuiso *et al* [3], Davison [4] and Nister [13]. In both categories, estimation of camera pose is based either on stepwise optimisation [9, 7, 13] or recursive filtering in the form of the Kalman filter [3, 4].

Although the above systems have demonstrated the potential of camera localisation, it is also true that they remain essentially laboratory based implementations. General purpose use in everyday applications is not yet achievable. If localisation is to move out of the laboratory then important limitations need to be addressed. Key amongst these are the related problems of (i) how to build in robustness to the often rapid and erratic motions associated with normal human activity, such as sudden movements and camera shake, and (ii) how to deal with the highly uncertain visual measurements resulting from the presence of dynamic clutter in the scene, such as moving objects, causing ambiguous feature matching, occlusion, etc. Existing methods rely critically on assumptions of static environments and either explicit or implicit models of Gaussian, uni-modal uncertainty. They consequently struggle when operating within such scenarios, where uncertainty is often multi-modal and rarely Gaussian.

The aim of the work reported here is to make progress on these issues. To do so, we have adopted a particle filtering approach. Particle filters provide general Bayesian, sequential estimation of state parameters using the principles of importance sampling, representing state posteriors by collections of random samples [1, 2]. This allows for the representation of non-Gaussian, multi-modal uncertainty in the state estimates, giving the potential for greater robustness when dealing with ambiguous or missing measurements, such as that discussed above. This has been used to good effect in other tracking tasks, including 2-D visual tracking [11] and robot navigation [12], and it has also been used for off-line structure from motion (SFM) [15]. In [14] we showed that similar gains can be made for real-time camera localisation. Several key advantages have emerged from using the approach. The ability of the filter to retain multiple modes for the state until confirmed or otherwise by subsequent measurements provides robustness against feature ambiguity and partial occlusion. At the extreme, under full occlusion for example, this results in 'particle spread', as the filter searches for profitable measurements to lock onto, giving it the ability to recover tracking, even after severe camera shake (and hence measurement loss). It also turns out to be easier to implement than existing methods and also scales better, with a linear increase in complexity with numbers of features, in contrast to the quadratic growth of the Kalman filter, for instance.

In the following we set out the basic methodology of the particle filter framework for camera localisation. Examples illustrating its use with known 3-D structure, in the form point depths and 3-D scene models, and unknown structure (SLAM) are then presented. The paper concludes with thoughts on the direction for future research.

2 Recursive Bayesian Camera Localisation

For camera localisation, we aim to recover the 3-D position and orientation of a moving camera at set time intervals with respect to a world coordinate frame. We set the latter to correspond to that of a reference camera frame, and represent the 3-D pose of the moving camera at time k by the state $\mathbf{x}_k = (\mathbf{q}_k, \mathbf{t}_k)$, where $\mathbf{t}_k$ denotes the camera position and the quarternion $\mathbf{q}_k$ encodes its 3-D orientation. The camera motion dynamics are assumed to be Markovian and defined by

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{v}_k) \tag{1}$$

where $f()$ is the potentially non-linear transition function and $\mathbf{v}_k$ is a noise component representing the uncertainty in the dynamics. For now, we also assume that knowledge of 3-D structure is available, which we take to be rigid and parameterised by a set of vectors $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$. In order to track the camera pose, we take measurements from each image frame captured by the camera, denoted $\mathbf{y}_k$, which relate both to the pose of the camera and to the structure parameters, via perspective projection. This we define using a general non-linear observation function $g()$ such that

$$\mathbf{y}_k = g(\mathbf{x}_k, Z, \mathbf{e}_k) \tag{2}$$

where $\mathbf{e}_k$ is a noise component representing the uncertainty in the measurements.

In a Bayesian context, localisation is then formulated as determining the posterior for the state, $p(\mathbf{x}_k|\mathbf{y}_{1:k}, Z)$, given the measurements and the structure, where $\mathbf{y}_{1:k} = \{\mathbf{y}_1 \ldots \mathbf{y}_k\}$ denotes the set of measurements obtained up until time k. Successive estimates can then be obtained, at least in theory, from the general form of the recursive Bayes filter [2]

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, Z) = \frac{p(\mathbf{y}_k|\mathbf{x}_k, Z)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{3}$$

which, up to a normalising constant, is defined by the likelihood of the latest measurement, $p(\mathbf{y}_k|\mathbf{x}_k, Z)$, derived from the observation relationship in (2), and the prior, $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$, which incorporates our belief about the camera motion. If the uncertainties in the transition and observation equations (1) and (2) are assumed to be Gaussian, then linearised variants of the Kalman filter algorithm, such as the extended or unscented Kalman filters, provide approximations to (3) [2, 16]. These represent successive state estimates and the associated uncertainties in terms of uni-modal means and covariances and provide very efficient implementations.

Although the Kalman approach can yield good localisation performance, as demonstrated by Davison [4] for example, the assumptions of Gaussian uncertainty present limitations in practice. The matching of features between frames is often ambiguous, due the nature of natural scenes, and this leads to multi-modal uncertainty in the measurements. In some cases this need not lead to tracking failure, particularly if the modes are covered by the Gaussian uncertainty and careful choice of measurements are made as in [4]. Also, if the camera motion corresponds closely to that assumed in the dynamical model (1), e.g. a constant velocity model, then this is likely to ensure that the filter continues to seek measurements in approximately the correct areas of subsequent frames, maintaining tracking through the ambiguities. However, if these constraints are violated, then tracking failure will result quickly; the Kalman filter will become unstable and diverge. This can happen when the camera undergoes rapid movement, such as shake, or when severe occlusion causes measurement loss, and both of these are common when operating in real applications.

The alternative is to seek a fuller representation of the posterior in (3) and this is provided by sequential Monte Carlo methods or particle filtering [1, 2]. These make no assumptions about the uncertainties in the state or measurements, and readily deal with the inherent non-linearities in the formulation. The principle is to approximate the posterior by a set of random samples and associated weights, thus enabling the representation of arbitrary densities with multiple modes. Ideally we would like to sample

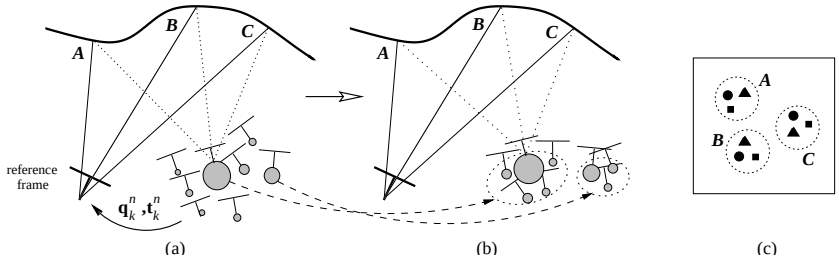


Fig. 1. Camera localisation using a particle filter: (a) each particle represents a potential 3-D pose for the camera, as defined by a quarternion $\mathbf{q}_k^n$ and a translation vector $\mathbf{t}_k^n$. Associated with each particle is a weight, indicated by the size of the filled circles, giving successive approximations to the posterior density for the motion parameters; (b) particles at the next iteration are populated around those with high weights in the previous iteration; (c) projections of the 3 points A, B and C into three different particles, indicated by the circle, triangle and square.

directly from the posterior in (3) but this is impractical; we do not have an analytical form and if we did it is unlikely that we could efficiently draw true samples from it [10]. However, we can get around the problem by adopting the principle of sequential importance sampling (SIS) [2]. This is a recursive process in which samples at a given iteration can be generated using an *importance density* from which samples can be easily drawn. A popular version of SIS which we adopt here is SIS with resampling, alternatively known as the Condensation algorithm [11]. Samples are generated at each iteration in proportion to the weights in the current sampled approximation to the posterior, hence populating new samples around potentially important modes, using the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as the importance density. Crucially, the weights for the new set of samples are then given by the measurement likelihood, $p(\mathbf{y}_k|\mathbf{x}_k)$, leading to a particularly straightforward algorithm, as described below.

3 Camera Localisation Using Particle Filtering

In a particle filter for camera localisation, assuming we have known structure, we need to sample within the 6 dimensional space of 3-D position and orientation, and each of the samples then corresponds to a potential camera pose. We refer to these samples as 'camera particles'. This is illustrated in Figure 1a, which shows particles at a given iteration, each corresponding to a possible pose for the camera, with their validity given by the sample weights, indicated by the size of the filled circles. For the next iteration, a new sample set is generated by first resampling according to the current estimate of the posterior (readers can refer to [2] for a description of resampling) and then evolving each sample into the next frame according the transition function in (1). The result is a higher population of new samples about those with greater weight in the previous iteration, as illustrated in Figure 1b. Weights for the new set of samples are then derived from a measurement likelihood, derived from the relationship in (2).

With known structure, computing the likelihood amounts to projecting the structure into each 'camera particle' and determining the closeness of the projection to measurements in the current frame. For example, given point structure as in Figure 1, the

weight for a particle is based on the similarity between regions around each projection and those around the (known) corresponding points in the reference frame. Note that in this case the projections into all camera particles form a set of 2-D 'particle clouds' in the current frame as illustrated in Figure 1c, which, as we show later, proves useful for minimising region similarity checks. Further iterations of the filter then proceed in a similar manner. Thus, given known structure, implementation of the filter is straightforward, requiring only a motion model (the transition function in (1)) and a measurement likelihood which can be readily evaluated for any given state. The form of these are discussed in the following sections.

There are several key advantages of this approach to camera localisation. In the presence of multi-modal measurement uncertainty, it has the ability to retain potentially significant modes until they are confirmed or otherwise by subsequent measurements. In the case of measurement loss or severe ambiguity then the sample weights will 'spread out', effectively widening the sample population across the state space, 'searching' for relevant measurements. As we show below, this plays a critical role in enabling the filter to avoid tracking failure even when faced with significant camera shake or severe occlusion.

4 Localisation with Known Point Structure

As noted above, implementation of the filter requires a motion model and a measurement likelihood. In this section we describe the form of these that we have used for the case when the location of a set of 3-D points in the scene are known. This is obviously a restricted class of localisation problems, but it serves to illustrate the performance characteristics of the particle filtering approach. We denote the set of particles and their associated weights at time k by $\{(\mathbf{x}_k^1, w_k^1), \ldots, (\mathbf{x}_k^N, w_k^N)\}$, where $\mathbf{x}_k^n$ is a particle defining a camera pose w.r.t the reference frame in terms of a rotation, encoded in the quarternion $\mathbf{q}_k^n$, and a translation, denoted by the vector $\mathbf{t}_k^n$, as illustrated in Figure 1a. The weight w_k^n is then proportional to the measurement likelihood $p(\mathbf{y}_k|\mathbf{x}_k^n, Z)$ and normalised s.t $\sum_{n=1}^{N} w_k^n = 1$. The structure is parameterised by the set of vectors $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_M\}$, where now each $\mathbf{z}_m$ denotes the 3-D location of the mth point. For each point, we define its projection into the nth particle as follows

$$\mathbf{u}(\mathbf{z}_m, \mathbf{x}_k^n) = \mathit{\Pi}(R_k^n \mathbf{z}_m + \mathbf{t}_k^n) \tag{4}$$

where R_k^n is the rotation matrix corresponding to the quarternion $\mathbf{q}_k^n$ and $\mathit{\Pi}$ denotes standard pin-hole projection for a calibrated camera (in our case, knowledge of the focal length).

4.1 Motion Model

At each iteration of the filter we evolve samples according to the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ as derived from the motion model in (1). This encodes our belief about how the camera pose is expected to evolve over time. A common approach when using the Kalman filter is to adopt a predictive motion model such as constant velocity [3, 4]. This has the

advantage of localising search regions for new measurements and hence minimising costly image processing operations. However, it also relies critically on camera motion being relatively smooth; deviations such as that caused by camera shake or rapid movement, can easily cause the filter to diverge, as noted in the previous section. In a particle filtering framework adopting predictive motion models can also be beneficial, in that it can significantly reduce the number of particles that need to be used; in effect, the motion model directs the particles into profitable areas of the state space, acting as an effective proposal density [2]. However, once again this assumes that the camera moves in a manner consistent with the model; if not, then recovery can be difficult for the particle filter when operating with fewer particles, limiting the potential for particle spread when measurement loss occurs. In our experiments with different localisation scenarios, we have found that a more effective approach to deal with rapid and erratic motion is to use a simple random walk model, based on a uniform density about the previous state, i.e.

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathrm{U}(\mathbf{x}_{k-1} - \mathbf{v}, \mathbf{x}_{k-1} + \mathbf{v}) \tag{5}$$

where $\mathbf{v}$ represents our uncertainty about the incremental motion. We therefore make the pragmatic assumption that we do not know how the camera pose is likely to evolve over time, and hence keep all possibilities open. We have adopted this form of motion model in the examples that follow.

4.2 Measurement Likelihood

The measurement likelihood $p(\mathbf{y}_k|\mathbf{x}_k^n, Z)$ needs to reflect our confidence in the camera particle $\mathbf{x}_k^n$. An obvious choice is to base it on the degree of correspondence between the projections $\mathbf{u}(\mathbf{z}_m, \mathbf{x}_k^n)$ and the projections of the points in Z into the reference frame. To do so, we use correlation. For each point in Z we record a template region in the reference frame and use this to generate a correlation field in the current frame. Camera particles giving projections close to high correlation values over many points in Z can then be regarded as having high likelihood. We have investigated two alternative ways of formulating such a likelihood. The obvious approach is to base it on the distance of a projection from the peak in the correlation field and this corresponds to the form of likelihood that is often used in Kalman filter techniques. The likelihood for particle $\mathbf{x}_k^n$ is then given by

$$p(\mathbf{y}_k|\mathbf{x}_k^n, Z) \;\propto\; \exp(-\sum_{m=1}^{M} (\mathbf{u}(\mathbf{z}_m, \mathbf{x}_k^n) - \hat{\mathbf{y}}_{km})^T C (\mathbf{u}(\mathbf{z}_m, \mathbf{x}_k^n) - \hat{\mathbf{y}}_{km})) \tag{6}$$

where $\hat{\mathbf{y}}_{km}$ denotes the peak position in the correlation field corresponding to point $\mathbf{z}_m$ and the covariance matrix C is diagonal and represents our uncertainty that the peak lies at the true corresponding point. The potential difficulty with this approach is that it implicitly assumes that the peak in the correlation is the best guess for correspondence and hence will favour camera particles which yield projections close to the peaks. This can cause problems if the peak is erroneous, as it often is for real data.

In the second method, we base the likelihood on how close a projected point is to any high correlation value, not necessarily the peak value. This removes the bias away

from a possibly erroneous peak and instead favours all particles which yield projections close to potential corresponding points. For this we adopt an inlier/outlier formulation as follows. Let $\mathbf{y}_{km}$ denote points in the current frame for which the correlation with the template associated with the point $\mathbf{z}_m$ is above a threshold. The likelihood is then given by

$$p(\mathbf{y}_k|\mathbf{x}_k^n, Z) \propto \exp(-\sum_{m=1}^{M} \prod_{\mathbf{u}\in\mathbf{y}_{km}} d(\mathbf{u}, \mathbf{z}_m, \mathbf{x}_k^n)) \tag{7}$$

where $d(\mathbf{u}, \mathbf{z}_i, \mathbf{x}_k^n)$ indicates whether the point $\mathbf{z}_m$ is an inlier ($d() = 0$) or an outlier ($d() = 1$) w.r.t the observation at $\mathbf{u}$ and the particle $\mathbf{x}_k^n$, i.e

$$d(\mathbf{u}, \mathbf{z}_m, \mathbf{x}_k^n) = \begin{cases} 1 \text{ if } ||\mathbf{u} - \mathbf{u}(\mathbf{z}_m, \mathbf{x}_k^n)|| > \epsilon_d \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

The threshold ϵ_d therefore defines a circular tolerance window around the projected point and the point becomes an inlier if a sufficiently high correlation value is within the window. We found that this form of likelihood gave better performance, particularly when dealing with erratic and rapid camera motions. An important point to note here is that although this likelihood is highly non-linear, it does not compromise the operation of the particle filter, as it would within a Kalman filter formulation. This illustrates another important advantage of the particle filtering approach in that it allows for greater flexibility in terms of both the observation and state transition models that can be adopted.

4.3 Particle Annealing

In the above likelihoods we need to determine suitable values for the covariance C and the tolerance window radius ϵ_d. There is a trade-off to address here. On the one hand, making them too small will make the likelihoods highly discriminatory, which in turn will result in a greater number of particles being required to provide sufficient coverage of the state space, which quickly becomes computationally prohibitive. On the other hand, making them too large will lead to reduced discrimination and hence localisation drift as particles become uniformly weighted. To address this we have adopted a multiresolution approach combined with particle annealing [8].

At each time step, we use a large value for ϵ_d (or, correspondingly, the diagonal elements of C) to obtain an initial set of weighted particles. Annealing then proceeds by resampling from this set using a uniform distribution with a smaller width than that used to propagate the samples from the previous time step ($\mathbf{v}$ in (5)). An updated set of weighted particles is then obtained using a smaller value of ϵ_d in the likelihood and this initialises the next annealing step. The process continues using smaller and smaller values of $\mathbf{v}$ and ϵ_d, resulting in greater concentration of the samples around modes in the state space. Further details on particle annealing can be found in [8].

4.4 Examples

An example of camera localisation using the above framework is shown in Figure 2. This shows successful camera localisation over an extended period as the camera is

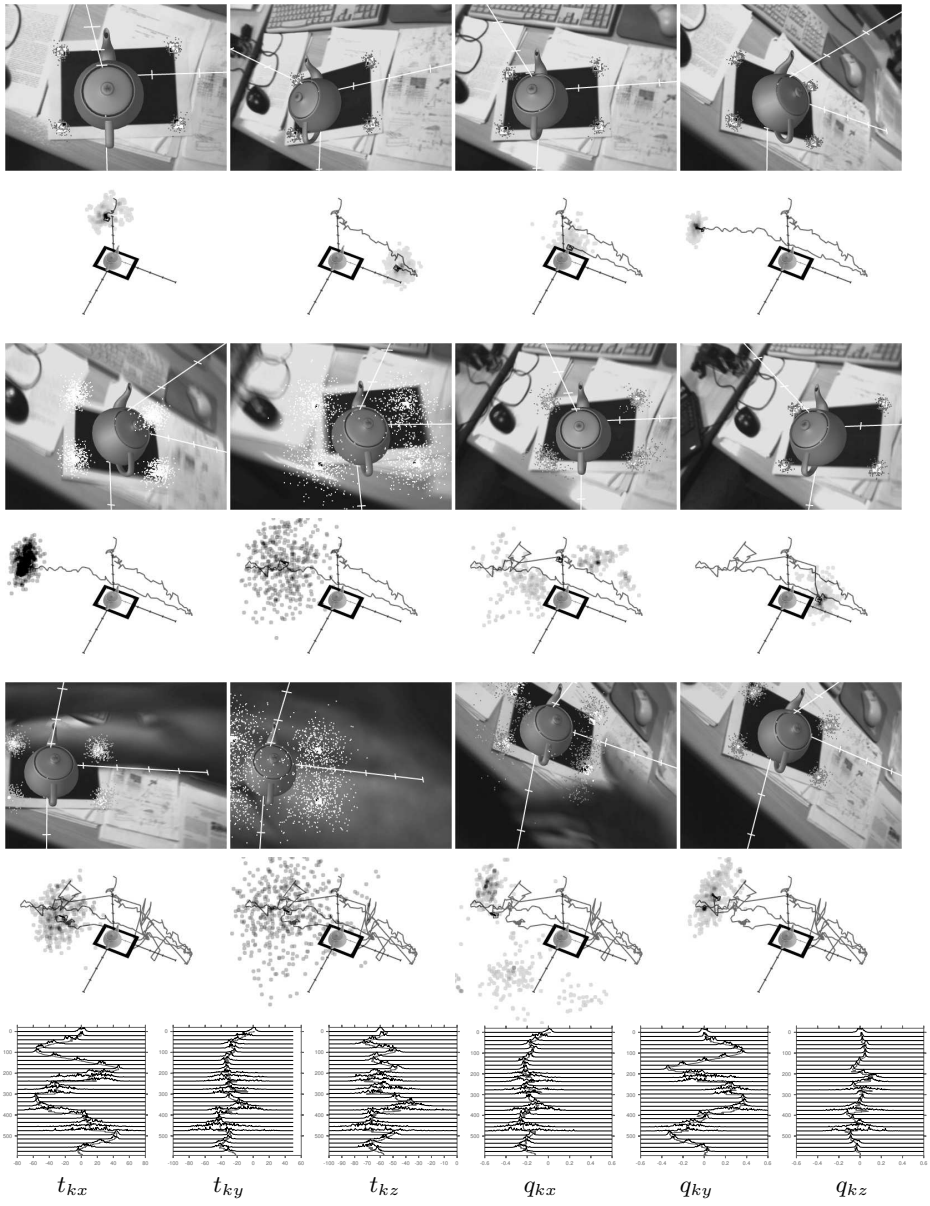


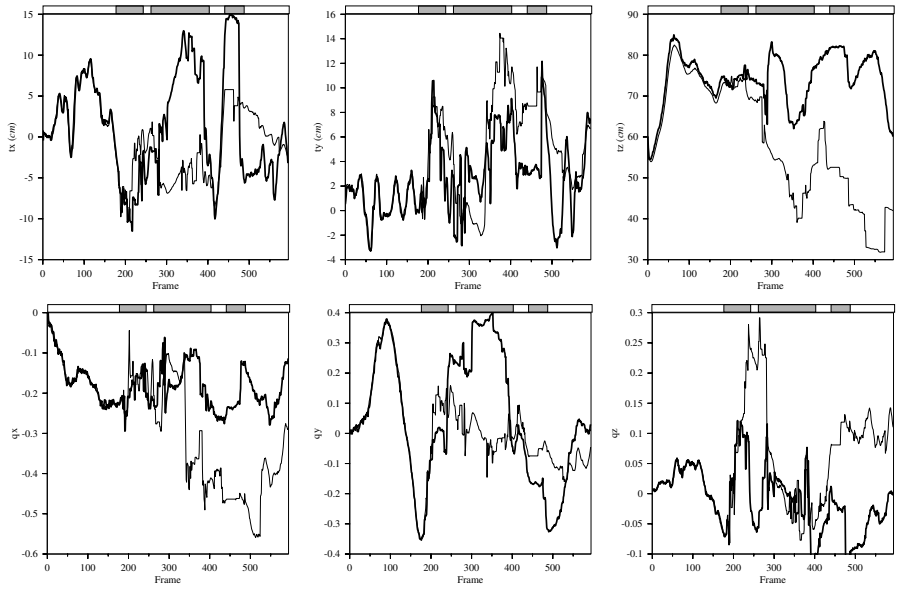
Fig. 2. Frames from camera localisation with known point structure. The views through the camera show augmented graphics synchronised with the video stream, indicating good camera localisation. Localisation continues even after rapid camera shake (rows 3 and 4) and after severe occlusion (rows 5 and 6).

moved with arbitrary motion over a desktop. We use a standard firewire web-cam with known focal length and the localisation is based on four known 3-D points in the scene corresponding to the corners of a rectangular test pattern. The filter was using 500

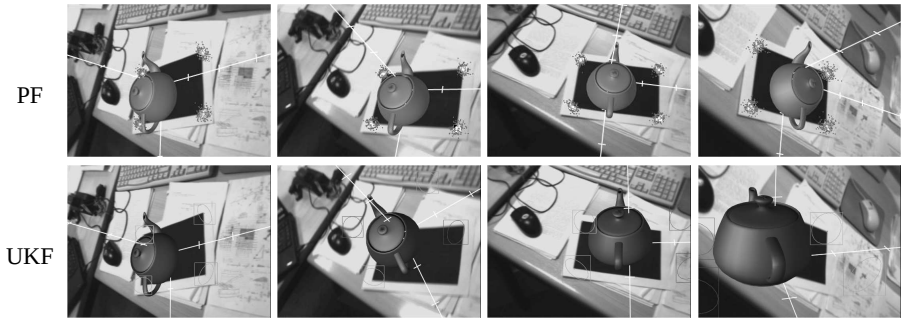
particles and the processing rate was around 40 fps. The top two rows of Figure 2 show the view through the camera and an external 3-D view showing the camera trajectory. The views through the camera have been augmented with a graphics object using the mean camera pose from the sampled posterior and this remains stable w.r.t the scene, indicating good localisation. Also shown are the projections of the 4 scene points into each frame for each of the camera particles. Note the concentrated distribution of the points, again indicating good localisation. The external views of the camera show the 3-D position of the particles for the given frame, where the darker points indicate particles with greater weight. The mean camera in the sample distribution is also shown.

The other rows in Figure 2 show results for when the camera is being rapidly shook (rows 3 and 4) and when the scene is severely occluded by a hand (rows 5 and 6). Readers should note the sudden spread of the particle distribution as shaking or occlusion occurs. This causes the filter to search wider over the state space until relevant measurements are locked onto and the filter stabilises, as shown in the images on the right-hand side. This can also be seen in the projections of the posterior for each of the motion parameters over the complete sequence shown at the bottom of the figure. Camera shake occurs between frames 180-250 and 320-400, and the occlusion occurs between frames 440-500. In each case the estimated posterior spreads and then converges once shaking or occlusion ceases. This illustrates the potential robustness of the particle filtering approach when faced with these types of camera motion.

To illustrate the benefits of the particle filtering approach, we have compared its performance on the above sequence with that obtained from a localisation algorithm based on the unscented Kalman filter (UKF) [16]. The framework for the UKF is similar to that for the particle filter, except that the camera pose is stored in terms of a mean and covariance, and the updates are based on the deterministic Kalman equations. We set the state and observation covariances to be comparable to the noise components within the particle filter in order to provide a fair comparison. The UKF measurements, one for each 3-D scene point, were taken as the distances from the peaks in the respective correlation fields. The camera pose estimates (position $\mathbf{t}_k$ and rotation quarternion $\mathbf{q}_k$) obtained from the two filters are shown in Figure 3a, where we have used the mean of the sample distribution from the particle filter (shown in bold). The shaded bars at the top of each plot indicate the frames in the sequence corresponding to the bouts of camera shake and occlusion. We do not have a ground-truth for the pose and so comparison is not straightforward. However, observation of the augmented camera views reveals that the UKF becomes unstable and diverges around frame 180, at the onset of the first bout of camera shake. After that the filter never recovers. In contrast, although the estimates from the particle filter are necessarily unreliable during shake and occlusion, the key point to note is that the filter recovers quickly once 'normal' camera motion resumes. This can be seen by comparing the augmented camera views for frames within the normal motion segments as shown in Figure 3b. For the particle filter, the augmentations are clearly synchronised with the video stream, indicating that the pose estimates are good; whilst those for the UKF are severely mis-aligned, indicating erroneous pose estimates. The degree of error can be appreciated from the divergence of the estimates from those of the particle filter in Figure 3a.



(a)



(b)

Fig. 3. Comparison of particle filter and UKF localisation algorithms: (a) pose estimates for the two filters, with those from the particle filter shown in bold (the shaded bars at the top of each plot indicate the bouts of camera shake and occlusion); (b) camera views with augmentation for selected frames during 'normal motion' segments.

5 Simultaneous Localisation and Mapping

Requiring knowledge of 3-D structure *a priori* severely limits the applicability of camera localisation. If vision based systems are to be useful then they need to operate easily over wide areas without pre-calibration. This can only be achieved if new scene structure is estimated (mapped) in tandem with estimating the camera pose. This is the simultaneous localisation and mapping (SLAM) problem for a single monocular camera

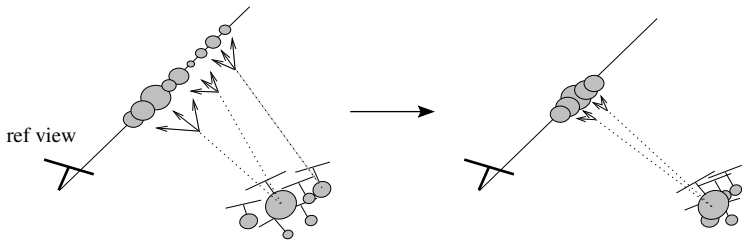


Fig. 4. In SLAM operation, depth estimates for new scene points are built up over time by triangulation of potential corresponding points over subsets of camera particles.

[4]. It is challenging problem, especially for real-time operation, since reliable depth information needs to mapped quickly so that it can be utilised for localisation. This involves careful selection of which new scene structure to map and effective mechanisms for incorporating the new structure into the localisation process.

Within a particle filtering framework, it is tempting to consider expanding the state space to include structure parameters and sample over these alongside the camera pose. This was the approach adopted by Qian and Chellappa [15] in their off-line SFM algorithm. Although they incorporated a partitioning scheme between sampling over pose and structure in order to counteract the problems inherent to increasing the state dimensionality, this quickly becomes computationally prohibitive for real-time operation, and does not represent a scalable solution. The alternative, and the one we have adopted, is to estimate scene structure outside of the main particle filter using an auxiliary process running in tandem with camera localisation. Once the structure estimates have converged then these can be incorporated into the scene map and utilised for localisation. This was the strategy adopted by Davison for a Kalman filter implementation of single camera SLAM [4] and our approach shares a number of similarities.

The basic idea is to build up depth estimates for a new point via triangulation based on the estimates of camera pose provided by the sampled approximations to the posterior. This works as follows. To insert a new point into the scene map, we detect salient points in a frame and record template regions about those points. The templates are then correlated with subsequent frames to indicate potential corresponding points. Since we have a density representation for the camera pose in both the initial (reference) frame and subsequent frames, we can triangulate potential corresponding points to build up a depth profile for each new point. Specifically, we assume that for the initial frame, the camera pose is given by the mean of the sampled posterior and that the new 3-D scene points then lie along rays passing through each salient point, as illustrated in Figure 4. In subsequent frames, for subsets of properly sampled camera particles, we triangulate each ray with points in the new frames having high correlation with the respective template and updating the weights in the vicinity of the triangulated depth based on the respective camera particle weight. Over time, we therefore build up a depth profile along each salient point ray. The process continues until either the weights converge on a given depth, and the point is then incorporated into the scene map, or the estimation is abandoned. New points are sought in each frame to ensure that sufficient numbers of active scene points are visible.

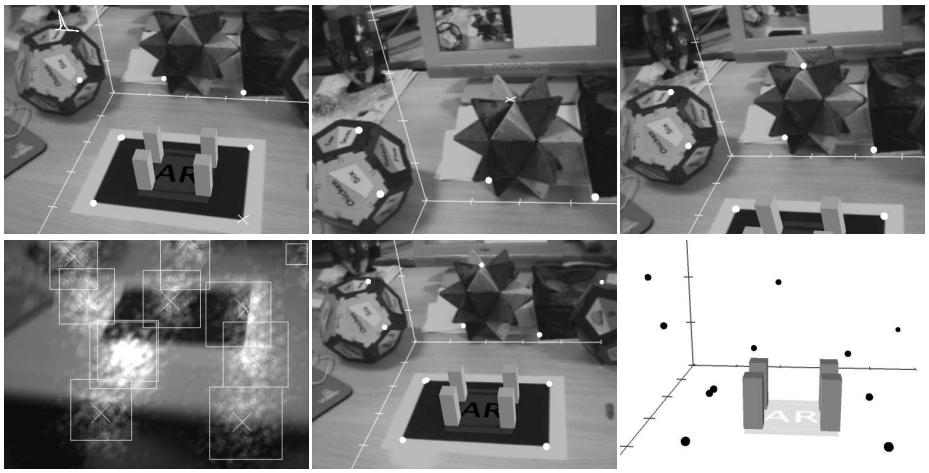


Fig. 5. An example of SLAM operation as the camera moves over a desktop environment, with successful localisation even when the test pattern moves out of view (top-middle image).

An example of SLAM in operation is shown in Figure 5. Again the system is initialised using the 4 corners of the rectangular test pattern. As the camera moves away, new points are incorporated as shown in the top-left image and localisation continues as the test pattern moves out of view. Eventually localisation is based only on the newly incorporated points as shown in the top centre image. Note that successful localisation continues even after severe shake (bottom-left image). The bottom-right image shows the estimated 3-D points in the scene map. As the camera returns to its original position the augmented graphics reappear and show reasonable stability, although some movement is also observed, indicating a drift in localisation as the camera moved away. This is caused by inaccuracies in estimating the 3-D positions of the new points, primarily due to our using the mean camera pose in the reference views, rather than proper sampling over the camera particles. We have recently developed an alternative method for initialising new points and we hope to report on this in the near future.

6 Camera Localisation Using 3-D Scene Models

An alternative to using known 3-D scene points is to base localisation on known 3-D models of objects in the scene. Such models can be built off-line and then imported into the filter for localisation purposes. This has been investigated before, see e.g. [5, 9], and can lead to increased stability and, crucially, greater view independence. The localisation based on scene points relies on matching templates, which become increasingly invalid over wide baselines, restricting the range of views for localisation. In contrast, having a full 3-D model allows prediction of the appearance of features in the scene for a given camera pose, hence giving view independent localisation.

Implementing this within a particle filtering framework is particularly straightforward. At each step of the filter and for each camera particle, we project the known 3-D

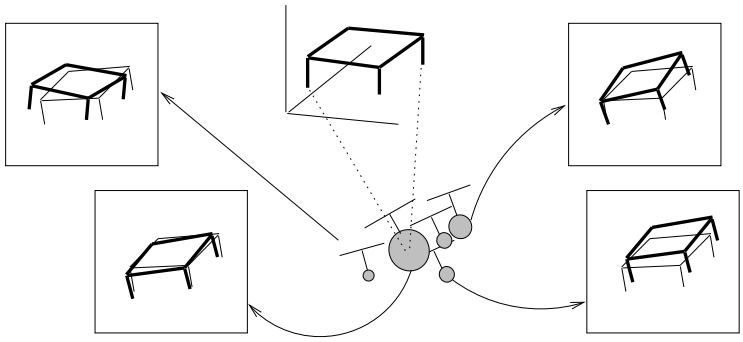


Fig. 6. Camera localisation using 3-D scene models. The measurement likelihood is computed by projecting the model into each camera particle and assessing the closeness of projected 3-D edges to those edges detected in the current frame. Particles yielding close agreement between projected and detected edges are then given greater weight.

model into the current frame. Measurements in the frame are then compared with their expected appearance from the projected model in order to derive the likelihood for the particle. This yields a set of weights in the usual manner and the filter proceeds as described earlier. For example, if the model is defined in terms of a set of 3-D edges, i.e a wireframe model, then the presence of 2-D edge features in the vicinity of their 2-D projection can be used for the likelihood. This is similar in form to the techniques used in 2-D contour tracking with particle filters [11]. The approach is illustrated in Figure 6. The key point here is the simplicity of the measurement model and also the fact that its inherent non-linearity does not compromise the operation of the particle filtering, as it would, say, if we were using a Kalman approach.

It turns out, however, that care does need to be taken when defining the measurement likelihood. We experimented with the straightforward approach of basing it on the sum of distances from discrete points on the projected model to the closest edge features detected in the frame. This proved less than successful, primarily due to the fact that the resulting likelihood lacks discrimination, causing the filter to quickly diverge in cluttered scenes. Essentially, all projections are able to find close edge features in such scenes and thus the weights in the filter disperse, leading to localisation failure. Our solution was to build a more discriminating likelihood based on key structures within the 3-D model. For this we used 3-D junctions and a likelihood defined in terms of how many junctions have sufficient support from features within the frame. We adopted a branch model for each junction and support for each branch is based on the closeness of edge features, as before. If all branches for a given junction have sufficient support, then the junction is deemed to be an inlier for the corresponding camera particle and the likelihood is then defined in term of the inlier/outlier ratio, in a similar manner to the likelihood used given known 3-D point structure.

An example of model-based localisation using a simple 3-D planar rectangle is shown in Figure 7. Here we have used the four corners of the rectangle as the key structures and modelled each as a junction with 2 branches. At each step of the filter, the branches are projected into the current frame for each camera particle (recall that each particle will yield a different projection, as illustrated in Figure 6). For a discrete

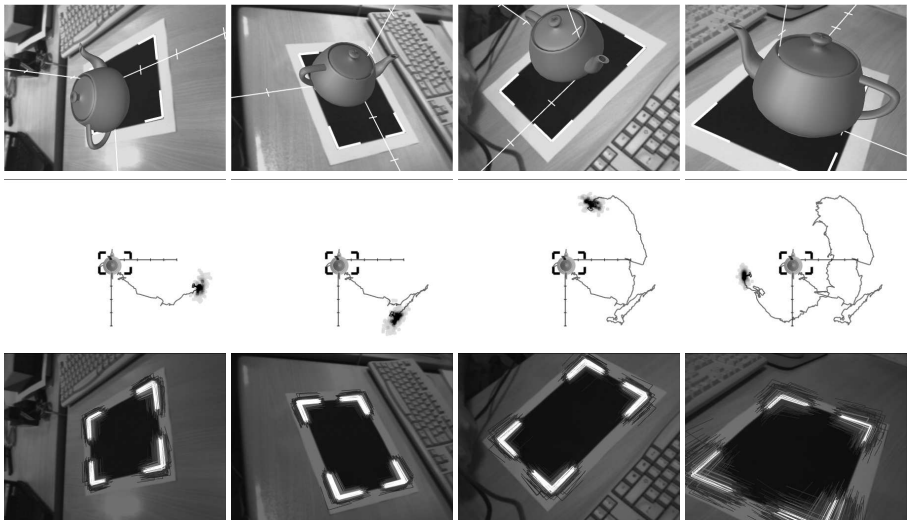


Fig. 7. Camera localisation using a known 3-D planar model.

set of points along each projected branch, we search along a perpendicular line and compute the distance to the closest edge feature detected in the frame. If such an edge feature is found within a threshold distance for all the points then the branch is deemed to be an inlier for that particle and if all branches are inliers then the junction itself is also an inlier. The likelihood for the particle is then derived from the number of inlier/outlier junctions, in a similar manner to that in (7). Despite the simplicity of this approach, localisation performance is good as can be seen from the results in Figure 7. The top two rows show the augmented camera views and 3-D external views, as before, whilst the bottom row shows the projected model for the set of camera particles, where the grey level indicates the particle weight. A key point to note is that now localisation can be achieved over a wide range of views (the camera actually circles the test pattern), which is in contrast to the limited range achievable using known point structure due to the view dependence of the template matching.

7 Conclusions and Future Work

We have summarised preliminary work on using particle filtering for real-time camera localisation. The aim is to seek greater robustness to the often rapid and erratic motions characteristic of normal human activity. This is necessary if camera localisation is to be transferred into areas such as wearable computing. The experiments suggest that the approach has considerable potential, demonstrating performance comparable to that of a standard Kalman filter during normal motion, whilst showing significantly better robustness to both erratic motion and severe visual occlusion. The key advantage that we have noted is the ability of the particle filter to recover localisation following such events in a natural and coherent manner, without the need for auxiliary re-initialisation.

There are, however, many avenues of further research that need to be explored. At present our template matching strategy for localisation with point structure is limited.

The templates quickly become invalid over wide baselines, restricting the range of operation. We are currently investigating methods to give greater invariance to view direction. The SLAM implementation described here also has its limitations, primarily due to the use of the mean camera particle when initialising new points. As noted earlier, we have recently developed a more effective initialisation method, which incorporates the uncertainty in the reference frame pose. We are also further developing the model based localisation framework, with a view to extending it to the use of full 3-D models, and also investigating other issues such as incorporating hidden line or surface removal. We aim to report on this and other aspects of the work in due course.

References

1. A.Doucet, N. de Freitas, and eds N.Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
2. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Trans on Signal Processing*, 50(2):174–188, 2002.
3. 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.
4. A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. Int Conf on Computer Vision*, 2003.
5. Tom Drummond and Roberto Cipolla. Real-time visual tracking of complex structures. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 24(7):932–946, 2002.
6. E.Foxlin and L.Naimark. Vis-tracker: A wearable vision-inertial self-tracker. In *Proc IEEE Virtual Reality Conference*, 2003.
7. Iryna Gordon and David G. Lowe. Scene modelling, recognition and tracking with invariant image features. In *Proc. Int Symp on Mixed and Augmented Reality*, 2004.
8. A. Blake J. Deutscher and I. Reid. Articulated body motion capture by annealed particle filtering. In *Proc Int Conf Computer Vision and Pattern Recognition*, 2000.
9. L.Vacchetti, V.Lepetit, and P.Fua. Stable real-time 3-d tracking using online and offline information. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 26(10), 2004.
10. D. J. C. MacKay. Introduction to Monte Carlo methods. In M. I. Jordan, editor, *Learning in Graphical Models*, NATO Science Series, pages 175–204. Kluwer Academic Press, 1998.
11. M.Isard and A.Blake. Condensation – conditional density propagation for visual tracking. *Int Journal of Computer Vision*, 29(1):5–28, 1998.
12. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. AAAI National Conference on Artificial Intelligence*, 2002.
13. D. Nistér. Preemptive ransac for live structure and motion estimation. In *Proc. Int Conf on Computer Vision*, 2003.
14. M Pupilli and A.D Calway. Real-time camera tracking using a particle filter. In *Proc British Machine Vision Conf*, 2005.
15. G. Qian and R. Chellappa. Structure from motion using sequential monte carlo methods. *Int Journal of Computer Vision*, 59:5–31, 2004.
16. 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.