Robust Tracking and Structure from Motion with Sample Based Uncertainty Representation

Peng Chang Martial Hebert

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{peng,hebert}@ri.cmu.edu

Abstract

Geometric reconstruction of the environment from images is critical in autonomous mapping and robot navigation. Geometric reconstruction involves feature tracking, i.e., locating corresponding image features in consecutive images, and structure from motion (SFM), i.e., recovering the 3-D structure of the environment from a set of correspondences between images. Although algorithms for feature tracking and structure from motion are well-established, their use in practical robot mobile applications is still difficult because of occluded features, non-smooth motion between frames, and ambiguous patterns in images. In this paper, we show how a sampling-based representation can be used in place of the traditional Gaussian representation of uncertainty. We show how sampling can be used for both feature tracking and SFM and we show how they are combined in this framework. The approach is exercised in the context of a mobile robot navigating through an outdoor environment with an omnidirectional camera.

1 Introduction

Geometric reconstruction of the environment from images is critical in mobile robot navigation. Geometric reconstruction involves feature tracking, i.e., locating corresponding image features in consecutive images, and structure from motion (SFM), i.e., recovering the 3-D structure of the environment from a set of correspondences between images. Although the basic algorithms for tracking and SFM are well understood, their operational use in the context of mobile robots in challenging conditions, including rough motion and complex 3-D shapes, remains difficult. In particular, occlusions, large change in motion of the robot, and noise in the images, all contribute to uncertainty in both the feature locations in the images and the 3-D structure. It is essential that the uncertainty be correctly modeled for the structure to be usable.

In this paper, we describe a sampling-based approach to represent the uncertainty in both tracking and SFM and integrate them into a single uncertainty maintenance algorithm. Our aim is to be able to apply standard tracking and SFM methods to situations in which Gaussian model would likely fail. These situations include complex environments in which features may be frequently occluded, robots operating in rough terrain, in which the smooth motion assumptions are not applicable.

2 Background

2.1 Problem Description and Notations

We assume that we are initially given a set of M features in a reference image I_o. We denote the position of feature j in I_o by z_o^j, $j = 1, \ldots, M$ and the vector containing the positions of all the features by $\mathbf{z}_o$. As the robot moves, new images are acquired, which we denote by $I_1, \ldots, I_k$. The initial features are located in the images using a feature tracker so that the location of feature j in image I_k is z_k^j and the vector of all the M feature locations is denoted by $\mathbf{z}_k$.

At time k, given $\mathbf{z}_o$ and $\mathbf{z}_k$, both the 3-D structure of the scene and the motion of the robot up to time k can be recovered. We denote by s_k^j the 3-D position of feature j reconstructed from $\mathbf{z}_k$ and by $\mathbf{s}_k$ the set of the 3-D coordinates of all M features. We denote by $\mathbf{m}_k$ the motion recovered at time k. Finally, we denote by $\mathbf{x}_k$ the pair $(\mathbf{s}_k, \mathbf{m}_k)$ reconstructed by SFM at time k.

The uncertainty on the locations z_k^j of the features in image k determines the uncertainty on the reconstructed structure and motion $\mathbf{x}_k$. Since $\mathbf{z}_k$ depends on the data in I_k and the positions predicted by the reconstruction $\mathbf{x}_{k-1}$, its uncertainty is described by the distribution $P(\mathbf{z}_k|\mathbf{x}_{k-1}, I_k)$. By maintaining this probability at each cycle, the tracking and SFM are integrated probabilistically. It can be shown that this uncertainty can be decomposed into a product of three terms, assuming $\mathbf{x}_{k-1}$ and I_k are conditionally independent given $\mathbf{z}_k$:

$$P(\mathbf{z}_k|\mathbf{x}_{k-1}, I_k) = K_1 P(\mathbf{z}_k|I_k) P(\mathbf{x}_{k-1}|\mathbf{z}_k), \qquad (1)$$

Equation 1 can be further converted to

$$P(\mathbf{z}_k|\mathbf{x}_{k-1}, I_k) = K_2 P(\mathbf{z}_k|I_k) P(\mathbf{z}_k|\mathbf{x}_{k-1}), \qquad (2)$$

where:

- $P(\mathbf{z}_k|I_k)$ is the uncertainty of the feature tracker alone;
- $P(\mathbf{z}_k|\mathbf{x}_{k-1})$ is the uncertainty obtained by transforming the structure computed at time $k-1$ with the predicted motion and then projecting the transformed structure into I_k [1];
- and K_2 is a normalizing constant involving only the priors. We will see that, because we use a sampled representation rather than a direct representation of the distribution, the normalization becomes unnecessary.

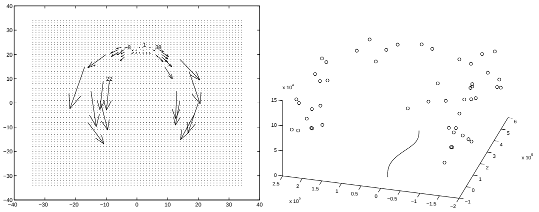
2.2 Gaussian Distributions vs. Sampled Distributions

Intuitively, Equation 2 provides a natural way to combine uncertainty in tracking and uncertainty in prediction from a noisy reconstruction from SFM. In principle, the uncertainty on the SFM at time k, described by the posterior distribution $P(\mathbf{x}_k|I_k)$, can be computed, given the uncertainty on $\mathbf{z}_k$,

In summary, we need to compute three crucial distributions, $P(\mathbf{z}_k|I_k)$, $P(\mathbf{z}_k|\mathbf{x}_{k-1})$, and $P(\mathbf{x}_k|I_k)$, in order to correctly represent the quality of the reconstruction at time k. Traditionally, these probabilities can be represented as Gaussian distributions. This is the approach taken in the approaches based on the Extended Kalman Filter (EKF) [1] [3] [4] [14] [15] [20]. However, given the fact that covariance representation is only a linear approximation to the uncertainty in the highly nonlinear SFM problem, the covariance representation is *not* a valid uncertainty representation for SFM in situations when (1) the correspondence noise is relatively large w.r.t. camera baseline, (2) correspondence noise can not be well approximated by a Gaussian distribution, or (3) the SFM result where the covariance is evaluated is not at the true minimum. Unfortunately these situations do occur in real navigation tasks respectively, when (1) some tracked features are far away from the robot, (2) there is ambiguity or several possible matches in tracking, i.e. the correspondence uncertainty has multiple modes, or (3) the SFM solution is a sub-optimal local minimum due to poor initialization.

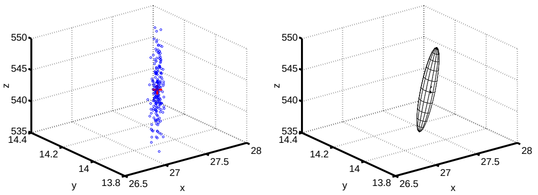
To illustrate these issues, consider a robot equipped with an omnidirectional camera moving along a specified path. Fifty features are tracked a two-frame SFM algorithm is used for recovering structure and motion. Details of the omnidirectional SFM algorithm can be found in [6]. The features are located at a range of up to 100m in front of the robot and 20m on the side. Figure 1 shows the observed optic flow on the omnidirectional image and the simulated environment.

We consider first a configuration in which the baseline between images is 19 meters and the uncertainty of the image locations of the features is Gaussian with variance 0.01 pixel. In that case, the uncertainty in the structure recovered by SFM is indeed well-approximated by a Gaussian distribution. Figure 2 shows the Monte Carlo runs of reconstruction for feature No.1, compared with the covariance representations at the true reconstructions for them.



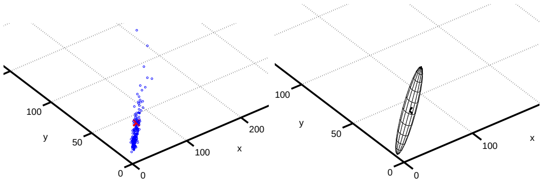
(a) (b)

Figure 1: (a) Optic flow observed from an omnidirectional camera (b) The simulated environment and robot motion



(a) (b)

Figure 2: (a) Samples of reconstruction of feature No.1 from Monte Carlo runs (b) Covariance approximation at true minimum



(a) (b)

Figure 3: (a) Samples of reconstruction of feature No.1 from Monte Carlo runs (b) Covariance approximation at true minimum

Since the covariance matrix is only a first order approximation of the true uncertainty, it cannot fully capture the uncertainty when the noise is large, however, as illustrated by

[1] Note that, to simplify the presentation, we use a simple constant motion model to compute $P(\mathbf{z}_k|\mathbf{x}_{k-1})$ but a more general dynamic model can be included in this framework

increasing the variance of the Gaussian noise to 1 pixel. Figure 3 shows that for the distant point (feature No.1) which has relatively small flow magnitude, the distribution becomes long-tailed, which cannot be approximated by Gaussian distributions. We further demonstrate the effect by projecting the reconstructions onto the main axis. We expect the distribution to be a Gaussian if the reconstruction can be approximated by covariance matrix. In fact we observe in Figure 4 a distribution with long tail for distant points, which is an indication that the underlying uncertainty cannot be captured by a Gaussian.

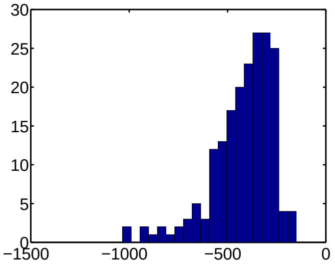
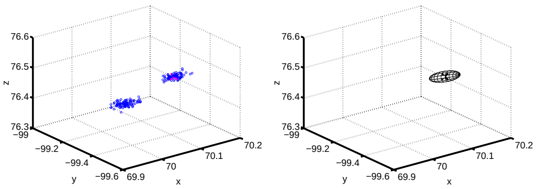


Figure 4: Histogram of the projections of the reconstructions for feature No.1

The tracker may return several possible correspondences for one feature. In such cases, the uncertainty becomes a multi-modal distribution. Figure 5(a) shows that the distribution of the reconstruction becomes multi-modal as well. In these situations, the simple covariance representation certainly would fail. Figure 5(b) shows that the covariance approximation can only capture one of the multiple modes.



(a) (b)

Figure 5: (a) Samples of reconstruction of feature No.22 from Monte Carlo runs (b) Covariance approximation at true minimum of feature No.22

To capture the uncertainty in situations in which the Gaussian model is insufficient, we propose to use a sampling method to represent the SFM uncertainty. Sampling methods provide a general framework to estimate the distribution of an estimator. Let us denote by $\hat{\theta} = g(t_1, \ldots, t_p)$ the estimator of an unknown p-dimensional value θ. If we know the distribution $P(t_1, \ldots, t_p)$, we can sample from them and form a set T of samples $(t_{1(i)}, \ldots, t_{p(i)})$, $i = 1, ..., N^2$. The estimator is applied to each sample, yielding estimates $\hat{\theta}_{(i)}, i = 1, \ldots, N$. The sample set $(\hat{\theta}_{(1)}, \ldots, \hat{\theta}_{(N)})$ is the representation of the uncertainty of $\hat{\theta}$. Monte Carlo (MC) stochastic algorithms have received much attention and sampling based non-parametric uncertainty representation have become popular, owing in large part to the increase in computational power. For example, successful systems have been demonstrated in the context of robot localization [7] and object tracking [11] [19]. Forsyth etc. [8] applied MC to SFM. Recently Qian and Chellapa [17] applied sequential MC methods to SFM problem to account for the non-Gaussian distributions in SFM results. But so far all the proposed MC based SFM algorithms can not account for the non-Gaussian distributions in tracking results (correspondences), therefore would not deal with the difficulties in real navigation tasks. In this paper, the uncertainties in both tracking and SFM are represented with sampling methods, and the proposed algorithm seamlessly integrates the tracking and SFM together to cope with the difficulties in real situations mentioned above. We explain how sampling techniques can be applied to represent the three probability distributions introduced above and to derive an estimator of $\mathbf{x}_k$.

2.3 Integrating Tracking and SFM

Feature tracking and SFM are often treated as separate problems with some notable exceptions. Direct approaches [2] [9] recover the camera motion without explicit feature correspondences. However direct methods assume small camera motion between frames which is not always true in robot navigation tasks. Torr [21] estimates the fundamental matrix with RANSAC and uses the recovered fundamental matrix to guide the feature matching. RANSAC implicitly builds an uncertainty model for the fundamental matrix, but it is only used for outliers rejection and the uncertainty is not propagated through time. In contrast, we build a complete uncertainty model for both tracking and SFM and interleave them together, e.g., through Equation 2, in a probabilistic way. Data association methods such as JPDAF can be effective in multiple feature tracking [18]. It has the advantage of holding multiple hypothesis, but how to incorporate geometric constraint from SFM is not immediately clear. Within our sampling-based probabilistic framework, multiple hypothesis are being tracked in a natural and principled way.

3 Sampling-Based Uncertainty Representation

3.1 Tracking Uncertainty

We use a standard feature tracker based on affine template matching [10]. This tracker computes the image location z at which the SSD error between the current image and an affine-warped version of a template from the previous image reaches a minimum. To simplify notations, we denote the difference between reference template and

[2] In an attempt to keep the flurry of indices under control, we always denote the sample number as a subscript in parentheses.

warped template at location z by $SSD(z)$. Assuming that the difference between the template and the image is caused by Gaussian noise, the distribution of the location z can be defined as [16]:

$$P(z) = exp(-kSSD(z)) \qquad (3)$$

where k is a normalizing scale chosen such that $P(z)$ integrates to 1. To represent the uncertainty in the tracker, a set of N sample locations is drawn according to the distribution of Equation 3. Given a set of M features, we denote by Z a set of samples drawn from the distribution, with $\mathbf{z}_{(i)}$ denoting the i-th sample, and $z^j_{(i)}$ denoting the position of the j-th feature from the i-th sample. More precisely, $\mathbf{z}_{(i)} = (z^1_{(i)}, \ldots, z^M_{(i)})$, $i = 1, \ldots, N$ is drawn from the combined distribution $P(z^1, ..., z^M) = \prod_{j=1}^{j=M} P(z^j)$, where $P(z^j), j = 1, \ldots, M$ is the distribution of Equation 3 for feature j.

Figure 6 shows a typical feature with the matched image region. It also shows the SSD surface and the samples from the distribution described by Equation 3.

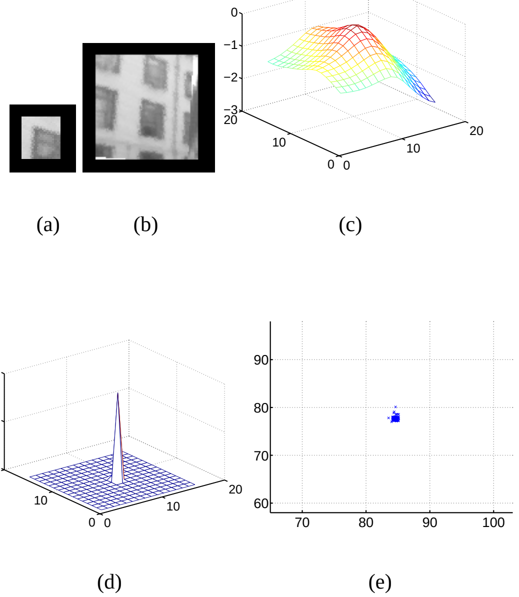


Figure 6: (a) one selected feature (corner) (b) the search region (c) SSD surface of the matching (The negative of the SSD surface is shown here to make the peaks more visible) (d) the density of the distribution according to Equation 3 (e) actual samples

Figure 7 shows a situation in which the location of the feature is ambiguous. In such cases, the uncertainty distribution 3 is multi-modal and cannot be represented by a Gaussian distribution.

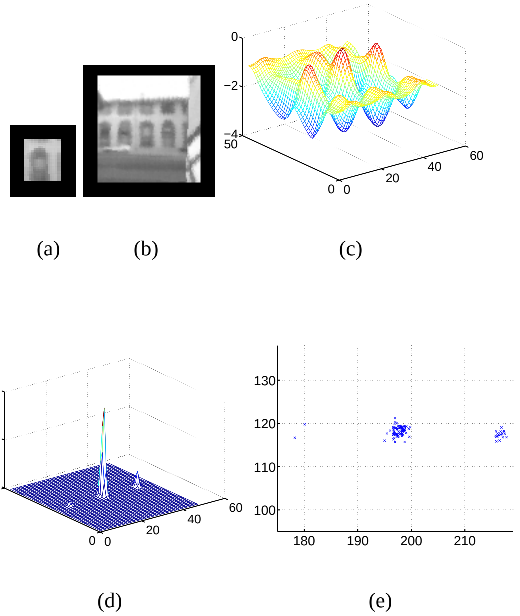


Figure 7: (a) selected feature (b) the search region (c) SSD surface of the matching (The negative of the SSD surface is shown here for same reason as before) (d) the density of the distribution according to Equation 3 (e) actual samples

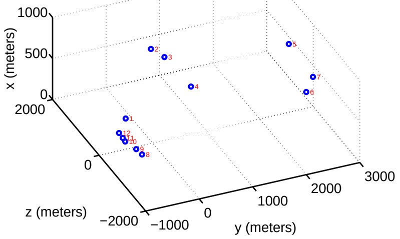


Figure 8: Reconstruction at true minimum

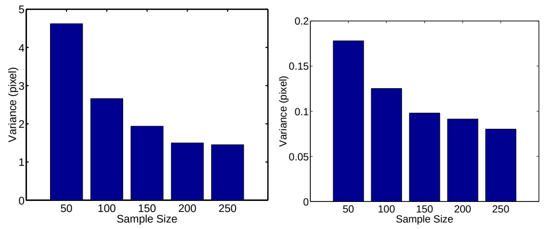
3.2 SFM Uncertainty

The uncertainty on structure and motion is represented by a set of samples $X = (\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(N)})$, in which each sample contains the structure and motion, $\mathbf{x}_{(l)} = (\mathbf{s}_{(l)}, \mathbf{m}_{(l)})$ computed from a sample of image feature locations $\mathbf{z}_{(l)}$ defined as in the previous section. Operationally, the SFM algorithms is executed N times, one for each sampled set of image locations, $\mathbf{z}_{(l)}$, $l = 1, \ldots, N$.

Figure 8 shows the structure samples of SFM, that is, for, each structure and motion pair $\mathbf{x}_{(i)} = (\mathbf{s}_{(i)}, \mathbf{m}_{(i)})$ generated from a sample $\mathbf{z}_{(i)}$, we display the M 3-D points in $\mathbf{s}_{(i)}$.

3.3 Sample Size

We have left the size N of the sample set unspecified so far. In fact, for the approach to be computationally tractable, it is important to verify that a modest sample size is sufficient. In the case of SFM with M features, the dimension of the space being sampled is $3M + 5$ (three coordinates per feature plus a rigid transformation up to a global scale factor.) Clearly, the sample size would be prohibitive if near-uniform coverage of the space were needed. In fact, a classical result due to D. McKay [13] shows that the accuracy of a Monte Carlo estimate depends on the variance of the estimated function but not directly on the dimensionality of the space sampled. In our case, it can be shown that a small number of samples is sufficient despite the high dimensionality of the space. As is common practice [11] [12] [19], we evaluate the sample size from training data. With the synthetic structure and given noise level similar to real situations, we determine the sample size required for the sampled estimate to reach a set level of accuracy. Since we are most interested in using the reconstructed 3D point distribution to guide the tracking, we compute the variance of the mean prediction for different sample sizes. As predicted by the theory, the variance decreases as the sample size increases. In practice, we choose a threshold of 2 pixels for the variance, which corresponds to a sample size $N = 200$. Figure 9 shows the variances decrease with increased sample size for distant features (Figure 9(a)) and nearby features (Figure 9(b)).



(a) (b)

Figure 9: (a) Variance of the mean reprojection v.s. sample size for feature No.1 (b) Variance of the mean reprojection v.s. sample size for feature No.22

4 Uncertainty Propagation

The basic issue issue in uncertainty propagation is: given an uncertainty representation of the structure and motion $\mathbf{x}_{k-1}$ at time $k-1$, and a new image I_k, compute the uncertainty on the new estimate of structure and motion, $P(\mathbf{x}_k|\mathbf{x}_{k-1}, I_k)$. A crucial aspect of the problem is that we need to explicitly combine the uncertainty on tracking and the uncertainty on SFM reconstruction. This is in contrast with most prior approaches in which the two sources of uncertainty are treated separately. The problem is further complicated by the fact that features may become occluded.

We describe first the core uncertainty propagation algorithm. Practical implementation issues and occlusion detection strategy are described in Sections 4.2 to 4.4.

4.1 Propagation Algorithm

0. Initialization: M features $\mathbf{z}_o = z_o^1, \ldots, z_o^M$ are selected in a reference image I_o and the corresponding features $\mathbf{z}_1 = z_1^1, \ldots, z_1^M$ are located in the second image I_1 using the affine feature tracker. A set Z_1 of N samples is drawn using the algorithm of Section 3.1. For each sample $\mathbf{z}_{1(i)}, i = 1, \ldots, N$, the corresponding structure/motion pair $\mathbf{x}_{1(i)}$ is computed. The sample set $X_1 = (\mathbf{x}_{1(1)}, \ldots, \mathbf{x}_{1(N)})$ is the representation of the initial uncertainty in scene structure and robot position.

Step 1. Estimate tracker uncertainty at time k ($P(\mathbf{z}_k|I_k)$): A set $Z' = (\mathbf{z}'_{(1)}, \ldots, \mathbf{z}'_{(N)})$ of samples of image locations is generated by using the result of the feature tracker in image k as shown in Section 3.1. Z' is a sampled representation of $P(\mathbf{z}_k|I_k)$.

Step 2. Propagate SFM uncertainty from time $k-1$ to time k ($P(\mathbf{z}_k|\mathbf{x}_{k-1})$): Let $\mathbf{x}_{k-1}$ be the structure reconstructed at time $k-1$. We assume that we have a sample set X_{k-1} representing the uncertainty on structure and motion at time $k-1$, $P(\mathbf{x}_{k-1}|I_{k-1})$. For each sample $\mathbf{x}_{k-1(i)}$, $i = 1, \ldots, N$, the corresponding set of 3-D points is transformed to image I_k using a motion model (a constant motion model in the simplest case), yielding a set of image locations $Z'' = (\mathbf{z}''_{(1)}, \ldots, \mathbf{z}''_{(N)})$. Z'' is a sampled representation of $P(\mathbf{z}_k|\mathbf{x}_{k-1})$.

Step 3. Combine tracker and propagated SFM uncertainty ($P(\mathbf{z}_k|\mathbf{x}_{k-1}, I_k) \propto P(\mathbf{z}_k|\mathbf{x}_{k-1})P(\mathbf{z}_k|I_k)$): The sample set Z', representing $P(\mathbf{z}_k|I_k)$, is resampled based on weights computed from the sample set Z'', representing $P(\mathbf{z}_k|\mathbf{x}_{k-1})$. The resulting new sample set Z is a fair sample of $P(\mathbf{z}_k|\mathbf{x}_{k-1}, I_k)$. The approach used for resampling - factored sampling - is described in detail in Section 4.2.

Step 4. Compute new SFM uncertainty at time k ($P(\mathbf{x}_k|\mathbf{x}_{k-1}, I_k)$): For each element $\mathbf{z}_{(i)}$, $i = 1, \ldots, N$ of Z, the corresponding structure $\mathbf{x}_{(i)}$ is computed. The resulting set $X_k = (\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(N)})$ is a sampled representation of the uncertainty on the reconstruction at time k, $P(\mathbf{x}_k|\mathbf{z}_k, \mathbf{x}_{k-1})$.

It can be shown that this sampled representation for $P(\mathbf{x}_k|\mathbf{z}_k, \mathbf{x}_{k-1})$ converges to the final uncertainty on reconstruction $P(\mathbf{x}_k|\mathbf{I}_k)$, where $\mathbf{I}_k$ represents all the images from time 0 to k.

4.2 Factored Sampling

Step 3. implements the relation $P(\mathbf{z}_k|\mathbf{x}_{k-1}, I_k) \propto P(\mathbf{z}_k|\mathbf{x}_{k-1})P(\mathbf{z}_k|I_k)$. Such a combination of sampled distribution can be achieved through “factored sampling” [11] for which a standard approach exists.

In factored sampling, if we weigh each sample in the sample set which represents $P(\mathbf{z}_k \mid I_k)$ by a weight proportional to $\mathbf{w} = P(\mathbf{z}_k \mid \mathbf{x}_{k-1})$, the resulting sample set will represent the conditional probability $P(\mathbf{z}_k \mid \mathbf{x}_{k-1}, I_k)$. The weights are estimated as follows: For every feature j and every sample ${z'}^j_{(i)}$ from Z', the weight $w^j_{(i)}$ is the number of sample points from Z'' that lie within a fixed radius of

$z'^{j}_{(i)}$. In practice, a radius of 2 pixels is used to compute the weights.

Once the weights are computed, the sample set Z' is resampled by using $w^j_{(i)}$ as the weight associated with each sample point. It can be shown that this weighted resampling procedure generates a fair sample Z of $P(\mathbf{z}_k|\mathbf{x}_{k-1}, I_k) \propto P(\mathbf{z}_k|\mathbf{x}_{k-1})P(\mathbf{z}_k|I_k)$ - see [11] for a justification of this approach to factored sampling and for details on the weighted resampling algorithms.

It is important to note that this procedure makes no assumption on the distribution of samples. In particular, the distribution is not required to be unimodal. Therefore, if there is an ambiguity in the tracking, e.g., two parts of the image are similar and closely spaced, the algorithm will preserve both alternatives in the sample set until such time that they can be discriminated.

4.3 Occlusion Detection

The algorithm is modified to include occlusions detection at *step 3*. When an occlusion does occur, the tracker would either (1) be unable to find any target within a search region or (2) find another feature with similar appearance to the tracked feature. Case (1) is relatively easy to detect by examining the SSD error or correlation value. Case (2) is considerably harder if no other information is provided. In traditional JPDAF-type approaches [18], a gating method is used, where the feature has to be within some distance to the predicted location. The actual threshold is decided by the assumed Gaussian covariance in measurement noise and system dynamics noise.

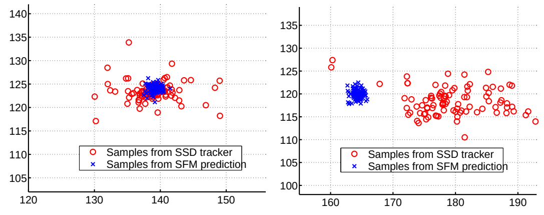
Occlusions are detected at *Step 3* of the algorithm, that is, after the resampling step described in the previous section. A given feature j is classified as occluded if the number of total number of samples from Z'' that fall within a 2-pixel neighborhood of a sample of Z' is lower than a threshold, that is, $\sum_{i=1}^{N} w^j_{(i)} < T$. T is a threshold that is currently set at $N/2$. It is worth noting that, in practice, the exact value of T is not critical to the performance of the algorithm.

It is important for features that are occluded to be allowed to "re-appear" at a later time. To allow this to happen, all the features currently occluded are examined after Step 4 for possible re-insertion in the list of visible features. If feature j is flagged as occluded at time $k-1$, then, at time k, it is projected to I_k using the estimate of the motion $\mathbf{m}_k$ (technically, the sample set of points representing feature j is transformed.) The tracker searches around this predicted location and a decision is made as to the visibility of the feature using the algorithm described above.

Figure 10 shows different situations in which occlusions occur and Figure 11 shows the effect of resampling.

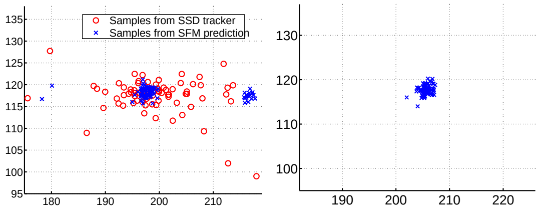
4.4 Sample Impoverishment

Sample impoverishment is a concern for any approach using sample to represent uncertainty [5]. This happens when the sample size is not large enough for the uncertainty in the system. The key difference between the approach described above and the conventional particle filtering approaches is that we sample from $P(\mathbf{x}_k \mid \mathbf{z}_k)$ at every time step, which means we effectively generate new samples for state variables at each time. In contrast, the usual particle filters only resample from current sample set thus no new samples are generated.



(a) (b)

Figure 10: (a) no occlusion: two sample sets are consistent (b) occlusion case (2): two sample sets are inconsistent



(a) (b)

Figure 11: (a) two sample sets before resampling (b) resampled sample set combining information from both tracker and SFM

5 Results

To illustrate the approach, we ran the the algorithm over sequences taken from a robot moving through a typical environment. To simulate the effect of occlusions, some of the images were edited to create artificial occlusion over several frames. As we see in the results, the system can (1) detect the occlusion when it happens, (2) guide the tracker to search the occluded target and (3) find the right target when there are ambiguities. This is difficult to achieved with EKF-based traditional approaches because (1) the uncertainty of the SFM involving remote features can not be captured by covariance representations thus accurate prediction is impossible when there is large motion, (2) when there is ambiguity (several possible locations) during the recovery, it cannot be represented by the covariance representation which assumes single-mode distributions.

Figure 12 shows the usual tracking result with prediction overlayed in frame No.3. The large quadrangles correspond to the search region used in the affine deformation. The small rectangles are the located features. Note that we are mostly interested in tracking feature No.1 (the one on the top middle view), which is the target the robot needs to go. The red dots are predicted samples from SFM. Without occlusion they are consistent with each other. Figure 13 shows the tracking result in frame No.8 where feature No.1 is occluded. The occlusion is detected (feature No.1 is not located in the figure) and the search region is enlarged. Figure 14 shows the tracking result with prediction overlayed in frame No.8. The search region is selected by the predicted samples from SFM. Even though the robot is undergoing a turning motion which causes a large translation of the feature in the image plane (more than 10 pixels between each frame), the predicted samples from SFM guide the search to the correct location of the occluded feature No.1. Figure 15 shows the tracking result with prediction overlayed in frame No.11. The search is still guided by the predicted samples from SFM correctly. Figure 16 shows the tracking result in frame No.13. The feature No.1 re-appears in the scene. Even though there are multiple objects similar to the original target due to the enlarged search region, the system is able to pick up the right one within several frames by combining information from both tracker and SFM probabilistically over time.

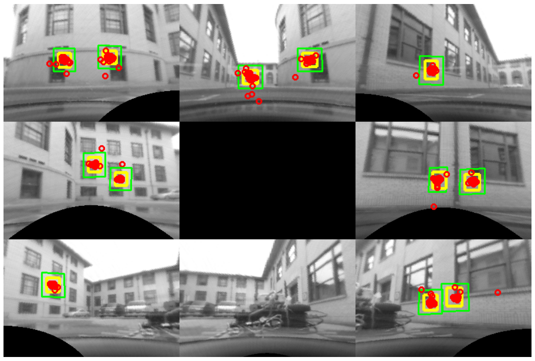


Figure 12: frame No.3: multiple feature tracking with prediction overlayed

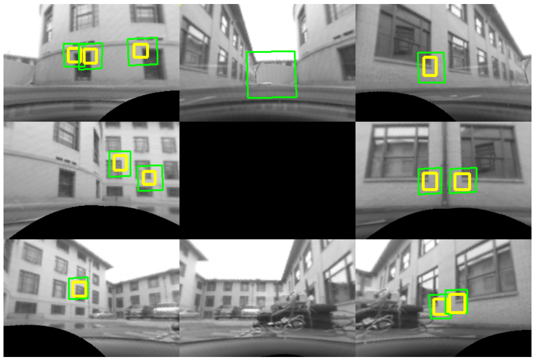


Figure 13: frame No.16: occlusion happens to feature No.1

To illustrate what the algorithm does, we show the sample distributions on image plane for feature No.1. Figures 10(a) to 11(b) show the evolution of the uncertainty distribution for feature 1. Figure 10(a) shows the samples from the tracker and SFM are consistent when there is no occlusion. Figure 10(b) shows the samples from the tracker and SFM are inconsistent when occlusion happens. Figure 11(a) shows when feature No.1 re-appears in the scene, the samples from the tracker indicate several possible locations, but the ambiguity is reduced with the samples from SFM as shown in Figure 11(b).

Figure 17 shows the distributions of recovered motion and structure parameters through time (16 frames total). Limited by space, only the first element of the quater-

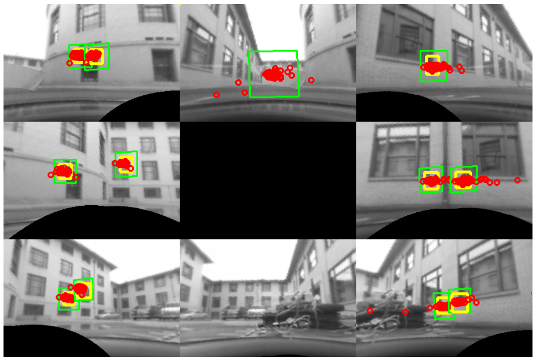


Figure 14: frame No.17: occlusion with prediction overlayed

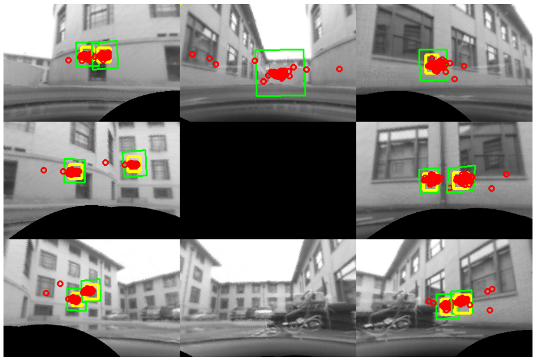


Figure 15: frame No.23: occlusion with prediction overlayed

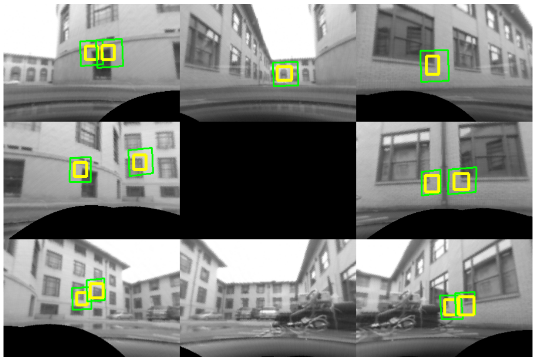
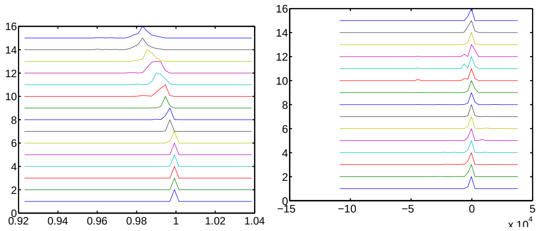


Figure 16: frame No.24: feature No.1 recaptured



(a) (b)

Figure 17: (a) Distribution of one motion parameter over time (b) Distribution of recovered X for feature No.1

nion and X coordinate of the structure for feature No.1 are shown. The distributions tend to be multi-modal at frame No.11 when there is confusion in tracking feature No.1, but they quickly concentrate again as new frames come in.

6 Conclusions

We have presented a sampling based method to characterize the uncertainty of tracking and SFM. It is able to capture the uncertainty in challenging situations in real robot navigation tasks in which the commonly used covariance representation would fail. We have also presented a sampling based filtering algorithm to propagate the uncertainty through time. Within our system the tracking and SFM are integrated probabilistically and the occlusions are handled in a principled way. The approach was validated in the context of a navigation task with an omnidirectional camera. The system exhibits robustness and improved tracking accuracy against occlusions.

We are currently conducting more careful evaluations for this algorithm in various navigation scenarios. Future work includes the combination with odometry sensor to better accommodate more dynamic robot motions, and the improvement on computational efficiency.

References

[1] A. Azarbayejani and A Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transaction on Pattern Recognition and Machine Intellegence*, (6), 1995.

[2] M.J. Black and P. Anandan. Robust dynamic motion estimation over time. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[3] J. Bouguet and P. Perona. Visual navigation using a single camera. In *International Conference on Computer Vision*, 1995.

[4] T.J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive estimation of 3-d kinematics and structure from a long image sequence. *IEEE Tran. AES*, (4), 1990.

[5] J. Carpenter, P. Clifford, and P. Fearnhead. Improved particle filter for non-linear problems. *IEEE Proc. Radar Sonar and Navigation*, (1), 1999.

[6] P. Chang and M. Hebert. Omnidirectional structure from motion. In *IEEE Workshop on Omnidirectional Vision, Hilton Head, SC*, 2000.

[7] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *International Conference on Robotics and Automation*, 1999.

[8] D.A. Forsyth, S. Loffe, and J. Haddon. Bayesian structure from motion. In *International Conference on Computer Vision*, 1999.

[9] B.K.P Horn and E. Weldon. Direct methods for recovering motion. *International Journal on Computer Vision*, (1), 1988.

[10] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.

[11] M. Isard and A. Blake. Condensation: conditional density propogation for visual tracking. *International Journal on Computer Vision*, (1), 1998.

[12] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *International Conference on Computer Vision*, pages 572–578, 1999.

[13] D. Mackay. Introduction to monte carlo methods. In *Learning in Graphical Models*. The MIT Press, 1999.

[14] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, (3), 1987.

[15] P. McLauchlan and D. Murray. A unifying framework for structure and motion recovery from image sequences. In *International Conference on Computer Vision*, 1995.

[16] K. Nickels and S. Hutchinson. Measurement error estimation for feature tracking. In *International Conference on Robotics and Automation*, 1999.

[17] G. Qian and R. Chellapa. Structure from motion using sequential monte carlo methods. In *International Conference on Computer Vision*, 2001.

[18] B. Rao. Data association methods for tracking systems. In *Active Vision*. The MIT Press, 1992.

[19] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conference on Computer Vision*, 2000.

[20] S. Soatto, R. Frezza, and P. Perona. Motion estimation on the essential manifold. In *European Conference on Computer Vision*, 1994.

[21] P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal on Computer Vision*, 24(3):271–300, 1997.