Scalable Monocular SLAM

Ethan Eade Tom Drummond
Cambridge University

{ee231, twd20}@cam.ac.uk

Abstract

Localization and mapping in unknown environments becomes more difficult as the complexity of the environment increases. With conventional techniques, the cost of maintaining estimates rises rapidly with the number of landmarks mapped. We present a monocular SLAM system that employs a particle filter and top-down search to allow real-time performance while mapping large numbers of landmarks. To our knowledge, we are the first to apply this FastSLAM-type particle filter to single-camera SLAM. We also introduce a novel partial initialization procedure that efficiently determines the depth of new landmarks. Moreover, we use information available in observations of new landmarks to improve camera pose estimates. Results show the system operating in real-time on a standard workstation while mapping hundreds of landmarks.

1. Introduction

Autonomous navigation in unknown environments requires knowledge of both pose and the environment's structure. This knowledge must be acquired online to make it useful. The process of causally estimating both egomotion and structure in an online system is *simultaneous localization and mapping* (SLAM), around which an impressive body of literature has been produced. Much of the research comes from the robotics community, where the direct application of SLAM to navigation is obvious. More recently, extensive work has been undertaken in computer vision to accomplish SLAM using visual data. This work is closely related to the structure-from-motion (SFM) problem of reconstructing scene geometry, and might be considered the proper subset of of SFM that uses causal or recursive estimation techniques.

The use of perspective-projection cameras as primary SLAM sensors introduces new difficulties to the problem. A single camera is a bearing-only sensor: it provides only two-dimensional measurements of three dimensional structure. Thus filtering methods that allow indirect observation models are crucial to the problem. The Kalman filtering framework that has emerged in the SLAM literature in the past two decades adequately supports the projective observation model. The Extended Kalman Filter (EKF) linearizes the observation and dynamics models of the system and represents all distributions as gaussians. A plurality of robotics SLAM systems employ the EKF. In the SFM literature, there has been significant success using the EKF for causal estimation – estimation depending only on observations up to the current time – with recursive algorithms [2, 4, 9]. In contrast to SFM approaches that rely on global nonlinear optimization, recursive estimation methods permit online operation, which is highly desirable for a SLAM system.

Davison shows the feasibility of real-time SLAM with a single camera in [5], using the well-established EKF estimation framework. His system takes a top-down Bayesian estimation approach, searching for landmarks in image regions constrained by estimate uncertainty instead of performing extensive bottom-up image processing and feature matching. Additionally, he describes a Bayesian partial-initialization scheme for incorporating new landmarks. However, while the performance of his system is accurate and robust, it cannot scale to large environments. The EKF maintains a full $N \times N$ covariance matrix for N landmarks, requiring $O(N^2)$ space. This covariance is updated with each measurement at $O(N^2)$ computation cost. These time and space requirements limit the total number of landmarks in the map to $\tilde{1}00$ if real time operation is desired. This level of map complexity allows localization and sparse mapping in a single room, but is not suited to very large areas or densely populated maps. Aggregated EKF updates[8, 10] allow efficient operation while observing a working set of landmarks, but full $O(N^2)$ updates are still required when changing the working set. Since the number of landmarks N grows with time, $O(N^2)$ update and storage costs quickly grow past a workable level.

This paper describes a SLAM system using a single camera as the only sensor, with the specific aim of frame-rate operation with many landmarks. Estimates are maintained in a FastSLAM-style[14] particle filter. To our knowledge, this is the first use of such an approach in a monocular SLAM setting, presenting significant challenges: The FastSLAM particle filter model must be reconciled with the top-

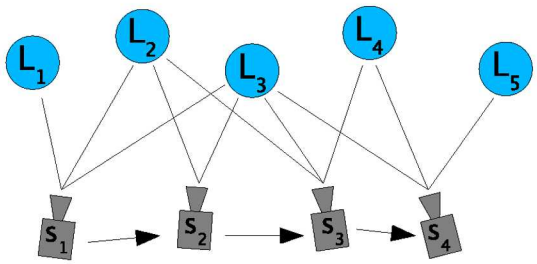


Figure 1. The vehicle or camera makes observations (shown as edges) of landmarks $\{L_i\}$ from each pose s_j. When the poses are known completely, the landmark estimates become independent, as landmarks are correlated only indirectly through their dependence on pose.

down active search required for efficient operation in a visual SLAM system. Additionally, the partial observations from a monocular sensor cause a high degree of coupling between landmark positions, and do not allow new landmarks to be simply incorporated into the system. To this end, we present an efficient algorithm for discovering the depth of new landmarks that avoids linearization errors. Finally, we describe a novel method for using partially initialized landmarks to help constrain camera pose.

2. Background

2.1. Scalable SLAM

Extensive work has been undertaken in the robotics community to address the complexity of large scale mapping[3, 7, 13, 23]. Several approaches explicitly model the weak covariance between geographically distant landmarks by fully decorrelating their estimates in submaps. In a submap of bounded complexity, computation and space requirements are also bounded. The EKF estimation framework can still be used in each submap, as long as a method exists for inferring migration from one submap to another. Other approaches enforce sparsity of correlation between landmarks in an adaptive manner, either by sparsifying inverse covariance [21] or by choosing a different representation of the covariance that is adaptively compressed [16].

More recently, Montemerlo et. al [14] have exploited a probabilistic property inherent to the SLAM problem: If the entire camera motion $\{s_i\}$ is known then the estimates of the positions of different landmarks become independent of each other. As shown in Figure 1, for any two landmarks L_i and L_j, and vehicle trajectory $\{s_i\}$,

$$p(L_i|L_j, \{s_i\}) \quad = \quad p(L_i|\{\mathbf{s}_i\}) \tag{1}$$

Each observation of a landmark is a measurement of the relationship between the vehicle and the landmark, so landmark estimates become correlated only when the vehicle position is uncertain.

To take advantage of this independence, Montemerlo et al. propose FastSLAM, which uses a particle filter to represent the distribution of vehicle poses. In each particle p_j, the landmark positions L_i^j are independent and can be represented analytically by distinct gaussians. Thus for N landmarks and M particles, FastSLAM requires $O(MN)$ space. Moreover, observation of a landmark requires updates to the pose estimates and only that landmark's estimate in each particle, at a total cost of $O(M)$. Each measurement causes the particles to be reweighted, and eventually the weights can converge to degenerate states. Resampling strategies avoid this diversity depletion, but require copying each particle's data. In FastSLAM, storing the landmark estimates in each particle in a balanced tree gives an update cost of $O(\log N)$ while allowing $O(M \log N)$ resampling. [14]. FastSLAM Particle filtering approximates a posterior distribution by sampling from a more tractable proposal distribution and weighting samples appropriately. Successful filtering requires high-quality proposal distributions, in the sense that they be as similar as possible to the desired posterior. In order to satisfy this criterion, proposal distributions should take into account the latest measurements available to the system [22]. FastSLAM 2.0 [15] shows how such improved proposals can be generated within the FastSLAM framework by adjusting the sampled poses according to new measurements. In order to successfully operate with few particles, this improved proposal generation scheme is crucial. If the transition prior given by the dynamic model is taken directly as the proposal, few or no pose samples will be sufficiently close to peaked observation likelihoods to accurately model the posterior. In vision-based systems such as the one described in this paper, observation likelihoods are extremely peaked relative to the broad transition prior given by the dynamic model. We employ a particle filter similar to FastSLAM 2.0 to maintain pose and landmark estimates in SLAM.

2.2. Vision SLAM with Particle Filters

FastSLAM has been previously applied to vision-based SLAM by Sim [20]. However, Sim's system uses a bottom-up approach to SLAM, building a large database of feature descriptors into which features from novel views are matched to localize the robot. This approach precludes real-time operation of his system, which has a mean processing time per frame of 11.9s. Furthermore, Sim's system uses a stereo camera rig, which simplifies the observation model but does not match the flexibility and low footprint of a monocular system.

Kwok and Dissanayake[11] use a modified particle filter to perform SLAM in a planar world by observing vertical edges with a camera. The system uses particle clouds to describe the probability distributions of landmarks in the world as well as robot pose. In contrast to FastSLAM,

this approach fails to take advantage of the probabilistic independence of landmarks given camera pose. Results are shown for only 33 landmarks, using 10,000 particles. The running time and scaling complexity of the system is not reported.

Pupilli and Calway[17] use a particle cloud to represent camera pose hypotheses, while landmarks are represented communally. The focus of the work is on robust camera localization, so results with many landmarks are not shown. Using 500 pose particles, the system operates at real time when observing four known landmarks, but drops to below frame rate when observing eight landmarks. Our goal is to map many landmarks in real-time while estimating camera motion accurately.

3. System Model

We use a single perspective-projection camera with known calibration parameters as a sensor. The state estimate of the system is encoded in a particle cloud. Each particle p_j, with associated weight w_j, corresponds to a complete pose and map hypothesis, consisting of a camera pose C_j and estimates of all landmarks $\{L_i\}$. Landmarks are observed in each video frame, and the particle cloud is updated to reflect the observations. The camera motion in each particle is governed by a dynamics model, and the observations by a measurement model, similar to SLAM in the EKF framework.

3.1. Dynamic Model

Each camera pose is an element of the Lie group of rigid Euclidean transformations, SE(3). Such an element C is stored as a transformation in projective three-space:

$$C = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{T} \\ \hline 0 & 1 \end{array}\right) \tag{2}$$

Camera velocities (in the camera frame) are elements of the tangent space, or Lie algebra, of SE(3). These are represented as six-dimensional vectors, with each dimension corresponding to a generator of the Lie group. Velocities are mapped onto geodesics in SE(3) with the exponential map $\exp$. To move a pose C by velocity μ over time δ, the pose is multiplied by the exponential of μ:

$$C_{t+\delta} = \exp(\delta\mu) \cdot C_t \tag{3}$$

For details of this representation, see [6, 9].

We assume a constant velocity motion model for each camera hypothesis, similar to that given in [5]. During each time step, a continuous random velocity walk with zero-mean white-noise acceleration occurs. This diagonal velocity noise covariance is given by

$$\mathbf{\Sigma} = \mathbf{diag}(\sigma_1^2, ..., \sigma_6^2) \tag{4}$$

Integrating this random walk in velocity over the time interval δ yields a 12-dimensional block-structured covariance $\mathbf{Q}$ of camera parameters and their velocities at the end of the time step:

$$\mathbf{Q} = \left(\begin{array}{c|c} \frac{\delta^3}{3}\mathbf{\Sigma} & \frac{\delta^2}{2}\mathbf{\Sigma} \\ \hline \frac{\delta^2}{2}\mathbf{\Sigma} & \delta\mathbf{\Sigma} \end{array}\right) \tag{5}$$

This process noise is used when incrementally improving the posterior as described in Sec. 4. The velocity components of each pose estimate are updated at the end of each time step according to the estimated pose. Velocity uncertainty in each camera pose hypothesis is not propagated from one time step to the next, as such uncertainty is encoded in the set of hypotheses as a whole.

3.2. Observation Model

A landmark is a three dimensional location along with a locally-planar image patch descriptor. For each particle, the gaussian estimate of a landmark L consists of a mean vector and covariance matrix in three dimensions:

$$L \equiv (\mathbf{x}, \mathbf{P}) \tag{6}$$

For a camera pose $C = (\mathbf{R}, \mathbf{t})$, the expected location $\mathbf{h}$ of landmark L in the camera plane is

$$\mathbf{h}(\mathbf{x}) = project(C \cdot \mathbf{x}) \tag{7}$$

$$C \cdot \mathbf{x} = \mathbf{R}\mathbf{x} + \mathbf{t} \tag{8}$$

$$project\left(\begin{pmatrix} x & y & z \end{pmatrix}^T\right) = \begin{pmatrix} x/z & y/z \end{pmatrix}^T \tag{9}$$

4. Recursive State Estimation

The FastSLAM 2.0 filtering framework has important differences from standard particle filtering methods. In particular, samples are not drawn until after observation updates, in order to take into account the latest observations. We describe the general operation below. For more details and proofs of correctness, see [15].

At each time step, three stages of computation take place: prediction, observation, and updates. In the prediction stage, a frame is captured from the camera, and the camera pose distribution is modified according to the dynamic model. At the end of the previous frame, the pose distribution was represented by a particle cloud. The linear dynamic model with process noise $\mathbf{Q}$ is applied to each particle's pose, yielding a gaussian distribution for each particle. Thus, the prediction stage turns the sampled representation of pose into a gaussian mixture representation of pose.

Using this predicted pose distribution, and the associated landmark estimates for each particle, landmark observations are extracted from the new frame (Sec. 4.1). The update stage then computes the posterior distribution by incorporating these observations (Sec. 4.2), and resamples poses.

Crucially, landmark estimates are not updated until new poses have been sampled, maintaining the independence of landmark estimates within each particle. Thus, at the end of processing for each frame, the distribution is again represented by pose samples with associated independent gaussian landmark estimates.

4.1. Top-down Observation Framework

In a general SLAM scenario, observations come from an abstract sensor, and the influx of observations does not depend on the estimation machinery. However, in a visual SLAM system using constrained active search for landmarks, this is not the case. Observations are made by actively searching new frames for landmarks. The search regions are determined by the current estimates of camera pose and landmark locations, and by the uncertainty in these estimates. In an EKF SLAM system like [5], the search region for a landmark is determined simply by projecting the expected landmark location into the image and projecting the landmark uncertainty by linearizing the observation model. However, with multiple pose hypotheses, and distinct landmark estimates for each pose hypothesis, a slightly different strategy for searching the image must be adopted.

For each landmark to be observed, the gaussian estimate of the landmark under each particle is projected into the image, by taking the weighted mean and covariance. This yields a single gaussian estimate of landmark location in the image. The corresponding 3σ ellipse in the image is searched for the landmark. The landmark's patch is warped by an affine homography $\mathbf{A}$ computed from the mode camera pose estimate and the initial camera pose from which the landmark's patch was captured. The location inside the ellipse yielding maximal normalized cross correlation (NCC) with the warped patch is taken as an observation of the landmark if the NCC score is above a threshold. If no such match is found, the landmark measurement is considered a failure. A simple heuristic based on the ratio of failed to successful measurement attempts determines when landmarks are removed from all particles' maps.

4.2. Refining the Posterior

The result of observing landmarks in a frame is a set of correspondences between image locations and landmarks. Each observation is assumed to have one-pixel measurement noise. These observations are used for two purposes: First, the gaussian mixture model of camera poses is refined according to the observations. Second, after poses are resampled from this updated distribution, the landmark estimates within each particle are updated using the same observations (and the newly-sampled poses). This allows the proposal distribution from which new poses are sampled to take into account the latest observations, while not destroying the conditional independence of landmark estimates inside each particle.

Given a set of observations, each observation can be used to incrementally update every component of the gaussian mixture model of poses. The update to each gaussian pose component is a combination of the prior (given initially by the dynamics model) and the likelihood (given by the observation), yielding a new gaussian posterior. The updates in this phase are identical to standard EKF updates[15]. The new particle weights $\{w_j\}$ are proportional to the likelihood of the set of observations given each particle j, obtained by integrating over camera pose, landmark position, and measurement noise. By linearizing the observation model, each weight can be computed in closed form. For a particle with landmark L (covariance $\mathbf{\Sigma}$), camera pose C, process noise $\mathbf{Q}$, observation jacobians $\mathbf{J}_L$ and $\mathbf{J}_C$, the set of observations $\mathbf{z}$ with predicted position $\hat{\mathbf{z}}$ and measurement noise $\mathbf{R}$ yields weight w:

$$w \quad \propto \quad \mathcal{N}\left[\hat{\mathbf{z}}; \mathbf{J}_C \mathbf{Q} \mathbf{J}_C^T + \mathbf{J}_L \mathbf{\Sigma} \mathbf{J}_L^T + \mathbf{R}\right](\mathbf{z}) \qquad (10)$$

All linearizations are computed about the initial mean of each pose gaussian (given by the prediction stage) so that the order in which observations are processed does not affect the cumulative result.

After all observations are processed, poses are randomly sampled from the gaussian mixture. Using standard resampling techniques[1], we create zero or more descendants from each particle according to its weight, with each descendant's pose sampled from the particle's associated gaussian. Data copying is minimized in the resampling operation by using copy-on-write: Each landmark estimate is copied only when a the estimate is modified (i.e., when the landmark is observed). Only shared pointers to landmark estimates need to be copied from a particle to its descendants during resampling. Using a tree, this can be further reduced from $O(N)$ to $O(\log N)$ time.

Once resampling has taken place, the same set of observations is then used again to update the estimates of the observed landmarks in each particle, with the standard EKF update equations. Because the landmark estimates within each particle are independent, each landmark update can be computed in constant time. Thus, at the end of each time step, the particle cloud is a set of samples drawn from the posterior distribution of poses and landmarks given by all observations up to the current time.

The total cost of updating landmark estimates and optimizing the proposal over M particles given k observations is $O(Mk)$, independent of the number of landmarks N. In contrast, the EKF with full covariance requires $O(N^2)$ time to make observation updates, which makes large numbers of landmarks impracticable.

5. Partial Initialization

In the above framework, landmarks are represented as three dimensional gaussians. With a single camera, the depth of a new landmark in the current view is unknown, and must be estimated from multiple views before a gaussian estimate of the landmark can be added to the map. Such a landmark is said to be *partially initialized*.

To this end, Davison maintains a set of depth hypotheses uniformly distributed along the viewing ray of a new landmark – a particle filter in one dimension[5]. Each observation is used to update the distribution of possible depths, until the distribution of depths is roughly gaussian, at which point the estimate is added to the map as a three-dimensional entity. Until this initialization occurs, the ray estimate is maintained in the system's single EKF. Lemaire et al. use a similar approach, but distribute depth hypotheses uniformly in inverse depth along the ray, as this corresponds to constant density of hypotheses when projected into the image[12]. As new measurements are made, Lemaire et al. repeatedly prune unlikely hypotheses until only one remains. A new landmark is initialized using the survivor hypothesis as a starting point.

5.1. Determining Landmark Depth

While both of the above techniques perform adequately as part of their respective SLAM systems, they are too expensive to maintain in our FastSLAM style system. With M particles, there must be M instances of the multiple-hypothesis depth filter for each new landmark. Observation updates of new landmarks then become expensive as the likelihoods of all hypotheses in all M instances must be evaluated. Furthermore, the depth range of new landmarks is limited by these approaches, as a hypothesis must exist with depth similar to that of the landmark to be initialized. With these concerns in mind, we propose a new partial initialization strategy suitable to our particle filter.

Instead of estimating the depth of new landmarks, we estimate the inverse depth in the frame of first observation. Consider a newly observed landmark, L_*, selected automatically from the image (we use the feature detector of [18]). In the camera frame from which it is first observed, let its three-dimensional location be given by

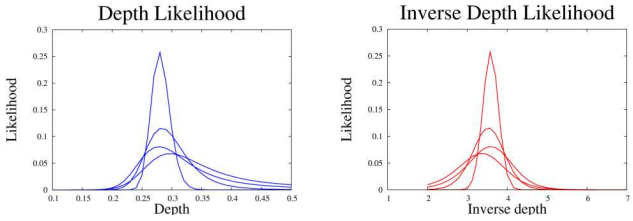
$$\mathbf{x}_* = \begin{pmatrix} x & y & z \end{pmatrix}^T \tag{11}$$

Instead of maintaining an estimate of these coordinates, we estimate the camera plane coordinates (u, v) and inverse depth q in this initial view:

$$\mathbf{b}_* = \begin{pmatrix} \frac{x}{z} & \frac{y}{z} & \frac{1}{z} \end{pmatrix}^T \tag{12}$$

$$= \begin{pmatrix} u & v & q \end{pmatrix}^T \tag{13}$$

As noted above, samples distributed uniformly in inverse depth along a viewing ray appear in novel views at uniform



(a) **(b)**

Figure 2. Estimating depth vs. estimating inverse depth, for several successive observations of a new landmark: In **(a)**, the estimates of depth likelihood converge to a gaussian-like shape, but the initial estimates are highly non-gaussian, with heavy tails. In contrast, likelihoods of inverse depth in **(b)** (abscissa is inverse meters) for the same data are nearly gaussian, even for the initial estimates. Thus linear techniques can be used to estimate inverse depth.

distribution along the epipolar line in the image. Consider the observation model under a simple camera displacement $\mathbf{t}$ (no rotation):

$$\mathbf{h}(\mathbf{x}_*) = project\left(\mathbf{x}_* + \mathbf{t}\right) \tag{14}$$

$$= \frac{1}{z + \mathbf{t}_z} \begin{pmatrix} x + \mathbf{t}_x & y + \mathbf{t}_y \end{pmatrix}^T \tag{15}$$

$$= \frac{z}{z + \mathbf{t}_z} \frac{1}{z} \begin{pmatrix} x + \mathbf{t}_x & y + \mathbf{t}_y \end{pmatrix}^T \tag{16}$$

$$= \frac{1}{1 + \mathbf{t}_z q} \begin{pmatrix} u + \mathbf{t}_x q & v + \mathbf{t}_y q \end{pmatrix}^T \tag{17}$$

When motion along the optical axis is small relative to the depth of the point ($\mathbf{t}_z q << 1$), this is very nearly a linear transformation in the modified coordinates.

$$\mathbf{h}(\mathbf{x}) \approx \begin{pmatrix} 1 & 0 & \mathbf{t}_x \\ 0 & 1 & \mathbf{t}_y \end{pmatrix} \begin{pmatrix} u \\ v \\ q \end{pmatrix} \tag{18}$$

This property implies that the EKF or its inverse form can be employed to estimate $\mathbf{b}_*$ in the local pose neighborhood of the first sighting of the landmark, as the distribution of the estimate is nearly gaussian. In contrast, the distribution of $\mathbf{x}_*$ is a cone with apex at the camera center and altitude along the optical axis, and a single gaussian approximation of this distribution is poor, as shown in Figure 2. We use linear Kalman filtering techniques to estimate $\mathbf{b}_*$ of a new landmark, independently in each particle's map. Each observation of the landmark is used to update the inverse depth estimate in each particle. When the uncertainty in inverse depth q is small enough that the distribution of the corresponding $\mathbf{x}_*$ is gaussian, the landmark is added to the particle's map as a fully initialized three-dimensional point. The change of variables is performed using the Unscented Transform [19], avoiding systematic bias that simply transforming the mean would induce.

One additional concern in partial initialization is that depth estimates of new landmarks converge artificially

when no new information is actually being observed regarding depth. For instance, if the camera is motionless, unmodeled correlations in observation noise cause the depth estimates of new landmarks to eventually converge to the mean scene depth of other observed landmarks. To counteract this early convergence, we discard any observations of new landmarks that lack sufficient depth information. This allows the camera to sit still without fully initializing new landmarks with spurious depth estimates.

5.2. Constraining Pose with Partially Initialized Landmarks

In existing SLAM systems, observations of partially initialized landmarks are not used to update estimates of camera pose. However, it should be noted that all observations of such landmarks in the image yield a two-dimensional measurement, which is used to estimate depth, a one dimensional quantity . This means that information in the measurement is being wasted. In our system, the rest of the information in the observation can be put to good use to improve camera pose estimates according to the epipolar constraint.

The location of a new landmark in the image can be considered in terms of the landmark's epipolar line, determined by the displacement from the first pose in which it was observed (C_0) to the current pose (C_1). The vector, in the camera plane, from the epipole (projection of C_0) to the observed landmark location has scalar components in the direction of the epipolar line and also perpendicular to the epipolar line. The first component yields information about the landmark's depth (or inverse depth). The second component should be zero for perfect estimates of camera pose.

Thus, the second component is a measure of epipolar reprojection error, and can be used in the filter in the same manner as observations of fully initialized points. Partially initialized landmarks provide one dimensional measurements of camera pose. The jacobians of this epipolar observation function are computed at the current estimate of $\mathbf{b}_*$ and employed, just like observations of fully initialized landmarks, to refine the posterior. Because these observations are effectively applying the epipolar constraint over multiple frames and a variety of frame pairs, they help constrain pose hypotheses in both rotation and translation, up to scale. Thus, even when viewing mostly partially initialized points (such as at the beginning of system operation) the camera poses can be well-constrained (Figure 9).

6. Results

Our SLAM system runs at frame-rate on a 2.8 GHz Pentium 4 workstation or 1.7 GHz Pentium M laptop while making 20-30 observations of landmarks in each frame. The processing time is nearly independent of the number of landmarks in the map, except for the very small but O(N) cost of deciding what landmarks to observe. Figure 3 shows that per-frame processing time does not increase noticeably with the number of landmarks.

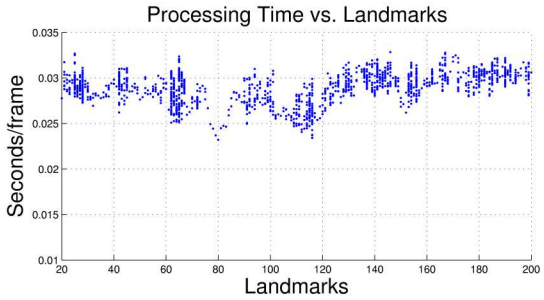


Figure 3. Time per frame vs. number of landmarks in the map. Processing time per frame is independent of the number of landmarks in the map.

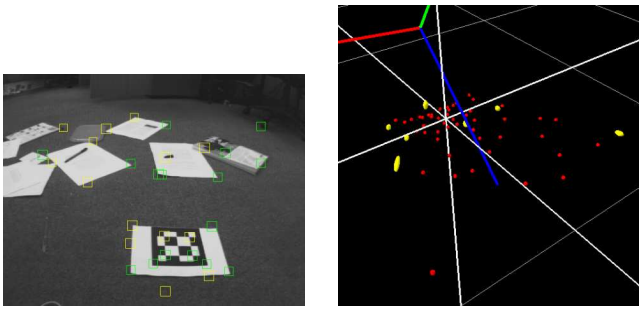


Figure 4. A view of a nearly planar scene, and the map generated from a 20 second motion. The landmarks cover a roughly 1m^2 area. The landmarks in the map, shown as 3σ ellipsoids, are coplanar to within 1 cm.

We show preliminary results of running our SLAM system in an indoor environment. The system is initialized by observing four points with known structure, giving it a measure of scale in the world. After initialization, all landmarks are acquired without user intervention. Figure 4 shows the map generated by observing many nearly-planar objects on a planar surface roughly one square meter in size, and one of the camera views of the scene. Landmarks in the map are represented by 3σ ellipsoids, except when the uncertainty in a landmark is very small, in which case it is drawn as a small but visible sphere. The generated map accurately reflects the planar structure, to within 1 cm. Some of this error can be attributed to the nonzero width of some of the objects.

Figure 5 shows the system traversing a loop. The camera starts at the initialization point, then moves away, and then returns by a different path, but reacquires the initialization points as it returns to the origin. Closing such loops is an important and difficult task for a SLAM system in order to minimize drift.

Figure 6 shows the map generated over a 3000 frame sequence. More than 250 landmarks are included in the map. Using 50 particles and making 20-30 observation updates

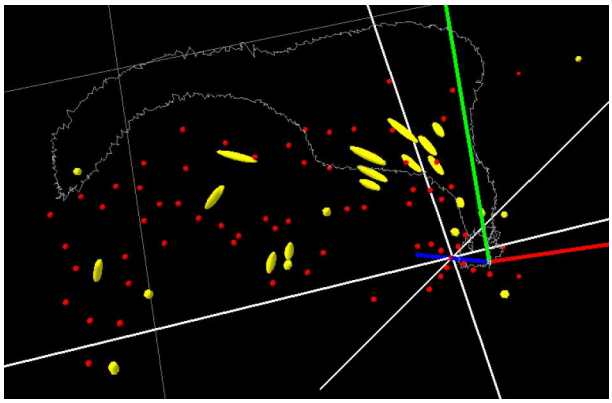


Figure 5. The camera, shown as coordinates axes, closes a loop. The trajectory trace is shown. There are 98 landmarks in the map.

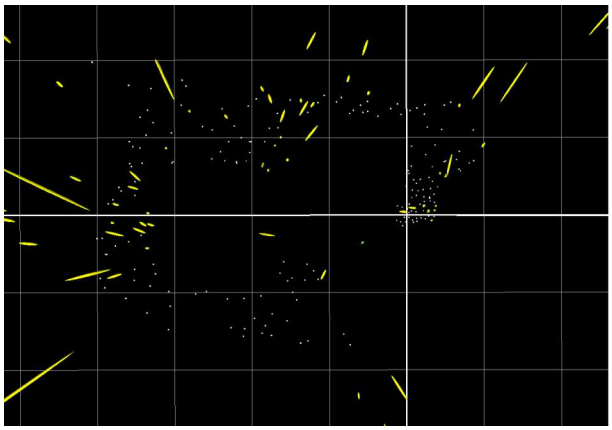


Figure 6. A map with 265 landmarks, viewed from overhead. The mapping was performed over a 100 second sequence at frame rate.

per frame, the system runs at frame-rate (30Hz) throughout the sequence. Because of the density of landmarks, the structure of the environment can be easily eyeballed from the mapping.

To evaluate the impact of the number of particles on the estimation, we use the same sequence with 50, 250, and 1000 particles. For all landmarks common to all particles at the end of the sequence, we generate from the particles a single large covariance matrix, with dimension $3N \times 3N$ for N landmarks. The eigenvalues of this matrix show how many dimensions of uncertainty are captured by the system. Specifically, we examine the off-diagonal covariance, because each particle maintains a diagonal estimate of landmark covariance: the distribution of particles must account for off-diagonal components. In all three cases the eigenvalues of the off-diagonal covariance drop to tiny values after the 30th eigenvalue, which implies that 50 particles is sufficient for the sequence. However, a more detailed and rigorous analysis will be necessary for longer and larger trajectories.

In Figure 7, new landmarks are being initialized into the map. The rays in the map on which the landmarks lie are shown in red. These rays correspond to the epipolar lines

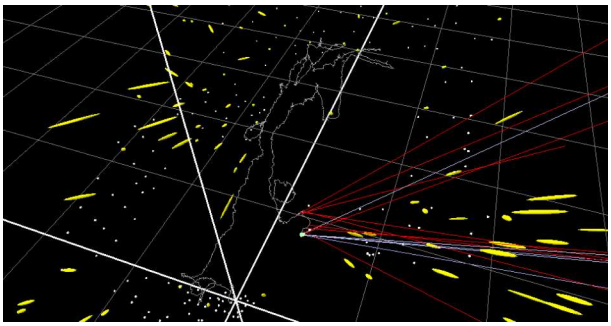


Figure 7. New landmarks being initialized. They lie on the rays shown in the map. When their inverse depth is determined with sufficient certainty, the rays become gaussian ellipsoids. The corresponding camera view is shown in Figure 8.

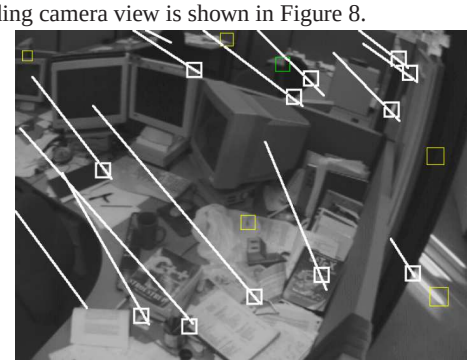


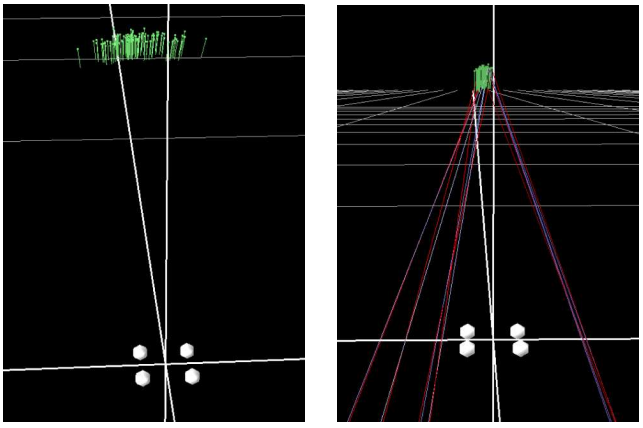
Figure 8. New landmarks being observed in the image. Epipolar lines, according to the mode particle's estimate, are overlaid. The distance of observed landmarks from their epipolar lines can help constrain pose: See Figure 9.

shown in Figure 8.

Partially initialized landmarks help constrain camera pose as described in Sec. 5.2. For instance, when the camera is stationary and viewing four planar points from a distance, there is an affine ambiguity that makes camera uncertainty high, as reflected by the particle cloud in Figure 9 (a). Because the camera is not moving, the depths of new landmarks cannot be determined. However, by enforcing the epipolar constraint, the pose can be considerably constrained, as shown in 9 (b).

7. Conclusions

We have presented a monocular SLAM system that operates at frame-rate while observing hundreds of landmarks. Our solution uses a FastSLAM particle filter to take advantage of the conditional independence inherent in the SLAM problem. We have shown how this filter can be incorporated into a single-camera, real-time system. Moreover, we have described a new partial initialization strategy for adding new landmarks to the map with a bearing-only sensor. This strategy estimates the inverse depth of new landmarks rather than their depths, and we have shown that this change of coordinates allows linearization algorithms to operate successfully.



(a) **(b)**

Figure 9. The camera pose estimate, shown at the top of **(a)** and **(b)** in particle-cloud form, is poorly constrained by observing only the four fully-initialized landmarks in **(a)**. By additionally applying the epipolar constraint to several partially initialized landmarks (no depth information), the camera pose is far better constrained, as shown in **(b)**.

There remain significant challenges to tackle with a particle-filter SLAM system intended to operate on large geographic scales. The number of particles necessary to maintain reasonable estimates of uncertainty in the pose and landmark estimates is not well known; it may increase with the environment complexity. While our system is capable of closing loops over short distances (though still with hundreds of landmarks), we have not yet evaluated its loop closing performance over larger trajectories. Because loop closing relies on accurate and consistent estimates of estimate uncertainties, adaptive particle sampling or an active loop closing algorithm may be necessary.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions of Signal Processing*, 50(2):174–188, February 2002.

[2] A. Azarbayejani and A. P. Pentland. Recursive estimation of motion, structure, and focal length. *Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[3] M. Bosse, P. Newman, J. Leonard, M.Soika, W. Felten, and S. Teller. An atlas framework for scalable mapping. In *ICRA*, pages 1899–1906, Taiwan, April 2003.

[4] Chiuso, Favaro, Jin, and Soatto. Structure from motion causally integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):523–535, April 2002.

[5] A. Davison. Real time simultaneous localisation and mapping with a single camera. In *ICCV*, Nice, France, July 2003.

[6] T. Drummond and R. Cipolla. Application of lie algebras to visual servoing. *Int. J. Comput. Vision*, 37(1):21–41, 2000.

[7] J. Guivant. *Efficient Simultaneous Localisation and Mapping in Large Environments*. PhD thesis, ACFR, Univ. of Sydney, Sydney, Australia, May 2002.

[8] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001.

[9] H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6):377–394, Oct 2003.

[10] J. Knight, A. Davison, and I. Reid. Towards constant time SLAM using postponement. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems, Maui, HI*, volume 1, pages 406–412. IEEE Computer Society Press, Oct. 2001.

[11] N. Kwok and G. Dissanayake. Bearing-only slam in indoor environments using a modified particle filter. In *Proc. Australasian Conference on Robotics and Automation*, 2003.

[12] T. Lemaire, S. Lacroix, and J. Sola. A practical 3d bearing-only slam algorithm. In *IROS 2005*, August 2005.

[13] J. Leonard and H. Feder. Decoupled stochastic mapping. *IEEE Journal of Ocean Engr*, 26(4):561–571, 2001.

[14] Montemerlo and Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *Proc. of IEEE Int'l Conference on Robotics and Automation*, Taipei, 2003.

[15] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *IJCAI*, pages 1151–1156, 2003.

[16] M. Paskin. Thin junction tree filters for simultaneous localizationand mapping. In M. Kaufmann, editor, *Proc. 18th IJCAI*, pages 1157–1163, San Francisco, CA, 2003.

[17] M. Pupilli and A. Calway. Real-time camera tracking using a particle filter. In *Proceedings of the British Machine Vision Conference*. BMVA Press, September 2005.

[18] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *ICCV 2005*, volume 2, pages 1508–1515, October 2005.

[19] J. K. U. S. J. Julier. A new extension of the kalman filter to nonlinear systems. In *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, pages 1628–1632, Orlando, Florida, USA, 1997. SPIE.

[20] R. Sim, P. Elinas, M. Griffin, and J. J. Little. Vision-based slam using the rao-blackwellised particle filter. In *IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, Edinburgh, Scotland, 2005.

[21] S. Thrun, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *IJRR*, 23(7):693–716, 2004.

[22] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical report, Cambridge University Engineering Department, May 2000.

[23] Z. Wang, S. Huang, and G. Dissanayake. Decoupling localization and mapping in slam using compact relative maps. In *Intl. Conf. on Intelligent Robotics and Systems*, pages 1041–1046, Edmunton, Canada, August 2005. IEEE/JRS.