σMCL: Monte-Carlo Localization for Mobile Robots with Stereo Vision

Pantelis Elinas and James J. Little
Computer Science Department
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
Email: {elinas,little}@cs.ubc.ca

***Abstract*— This paper presents Monte-Carlo localization (MCL) [1] with a mixture proposal distribution for mobile robots with stereo vision. We combine filtering with the Scale Invariant Feature Transform (SIFT) image descriptor to accurately and efficiently estimate the robot's location given a map of $3D$ point landmarks. Our approach completely decouples the motion model from the robot's mechanics and is general enough to solve for the unconstrained 6-degrees of freedom camera motion. We call our approach *σMCL*. Compared to other MCL approaches σMCL is more accurate, without requiring that the robot move large distances and make many measurements. More importantly our approach is not limited to robots constrained to planar motion. Its strength is derived from its robust vision-based motion and observation models. *σMCL* is general, robust, efficient and accurate, utilizing the best of Bayesian filtering, invariant image features and multiple view geometry techniques.**

I. Introduction

Global localization is the problem of a robot estimating its position by considering its motion and observations with respect to a previously learned map. Bayesian filtering is a general method applicable to this problem that recursively estimates the robot's belief about its current pose. Monte-Carlo localization provides an efficient method for representing and updating this belief using a set of weighted samples/particles. Previous MCL approaches have relied on the assumption that the robot traverses a planar world and use a motion model that is a function of the robot's odometric hardware. Based on uninformative sensor measurements, they suffer from the perceptual aliasing problem [2] requiring that the robot move for several meters and make many observations before its location can be established. They also demand a large number of particles in order to converge. MCL has been demonstrated to be accurate for laser-based robots but it has failed to achieve similar results for vision-based ones.

In this paper, we present Monte-Carlo localization for robots with stereo vision. We call it σMCL and it differs from others in several ways. Firstly, it is not limited to robots executing planar motion. We solve for unconstrained $3D$ motion (6 degrees of freedom) by decoupling the model from the robot's hardware. We derive an estimate of the robot's motion from visual measurements using stereo vision. Secondly, we use sparse maps of $3D$ natural landmarks based on the Scale Invariant Feature Transform [3] that is fully invariant to changes in image translation, scaling, rotation and partially invariant to illumination changes. The choice of SIFT leads to a reduction of perceptual aliasing enabling σMCL to converge quickly after the robot has traveled only a short distance. Finally, our method is more accurate than other constrained vision-based approaches and only requires a small number of particles.

In comparison, Thrun et al. [1] introduce MCL and study its performance for planar, laser guided robots that utilize $2D$ occupancy grid maps [4]. They also demonstrate it for a robot with vision that relies on an image-based mosaic of a building's ceiling but fail to match the accuracy of their laser-based approach [5]. Wolf et al. [6] implement MCL for a vision guided robot that uses an image retrieval system based on invariant features but also needs $2D$ occupancy grid maps for visibility computations. Their system requires the storage of a large database of images along with the metric maps.

Recently, image-based approaches have been proposed that do not require metric maps but operate using collections of reference images and their locations. These methods combine filtering with image-based localization [7] in a more general setting than originally proposed by [6]. Variations exist for different choices of image descriptors and associated similarity metrics. Menegatti et al. [8] represent images using the Fourier coefficients of their lower frequency components. They define a simple similarity metric using the Euclidean distance of these coefficients. Gross et al. [9], in a similar approach, use the Euclidean distance of the mean RGB values of images as a similarity metric and employ a luminance stabilization and color adaptation technique to improve matching accuracy. Ulrich et al. [10] represent images using color histograms in the normalized RGB and HLS color spaces. They study the performance of different similarity metrics for their histogram representation. Kröse et al. [11] perform Principle Component Analysis on images and store the first 20 components. They match images by comparing the Euclidean distance of these components smoothed by local Gaussian kernels. For improved efficiency, they implement an approximate nearest-neighbour approach using the kd-tree data structure. Rofer et al. [12] propose a variant that focuses on fast feature extraction. Theirs is limited to very small environments and it depends on color-based landmarks suitable only for the robot soccer domain.

The rest of this paper is structured as follows. We begin

with an overview of Bayesian filtering and its application to robot localization leading to MCL. We describe the acquisition of maps and continue to present the main elements of *σMCL*, namely its vision-based motion and observation models. We provide experimental results to prove its accuracy. We compare it with other vision-based MCL methods and show that it performs better. Finally, we conclude and suggest directions for future work.

II. Bayesian Filtering with Particle Filters

Our goal is to estimate the robot's position and orientation at time t, denoted by s_t. There are 3 degrees of freedom for the position (x,y and z) and 3 for the orientation ($roll$, $pitch$ and yaw). That is, s_t is a 6-dimensional vector. The state evolves according to $p(s_t|s_{t-1}, u_t)$ where u_t is a control signal most often an odometry measurement. Evidence, denoted by y_t, is conditionally independent given the state (Markov assumption) and distributed according to $p(y_t|s_t)$. Bayes filtering recursively estimates a probability density over the state space, given by [1]:

$$p(s_t|y^t, u^t) = Bel(s_t) = \alpha p(y_t|s_t) \int_{s_{t-1}} p(s_t|s_{t-1}, u_t) Bel(s_{t-1}) ds_{t-1} \quad (1)$$

It's performance depends on how accurately the transition model is known and how efficiently the observation probability density can be estimated.

Particle filtering is a method for approximating $Bel(s_t)$ using a set of m weighted particles, $Bel(s_t) = \{x^{(i)}, w^{(i)}\}_{i=1,\cdots,m}$. The system is initialized according to $p(s_0)$ and the recursive update of the Bayes filter proceeds in the following steps:

1) for each particle i
2) Sample from $Bel(s_{t-1})$ using the weighted samples, giving $s_{t-1}^{(i)}$
3) Sample from $q_t = p(s_t|s_{t-1}, u_t)$ (also known as the proposal distribution), giving $s_t^{(i)}$
4) Compute the importance weight, $w^{(i)}$ according to $p(y_t|s_t^{(i)})$
5) end for
6) Normalize the weights such that they add to 1.0
7) Resample from the particles proportionally to their weight

This procedure is known as *sampling-importance-resampling*. The application of the particle filter to robot global localization is know as MCL. It has been shown that the choice of proposal distribution is important with respect to the performance of the particle filter. An alternative proposal distribution that leads to what is called *dual MCL* is suggested in [13]. Dual MCL requires that we can sample poses from the observations using the dual proposal distribution (a hard problem in robotics)

$$\tilde{q}_t = \frac{p(y_t|\tilde{s}_t)}{\pi(y_t)}, \ \pi(y_t) = \sum_{\tilde{s}_t} p(y_t|\tilde{s}_t) \quad (2)$$

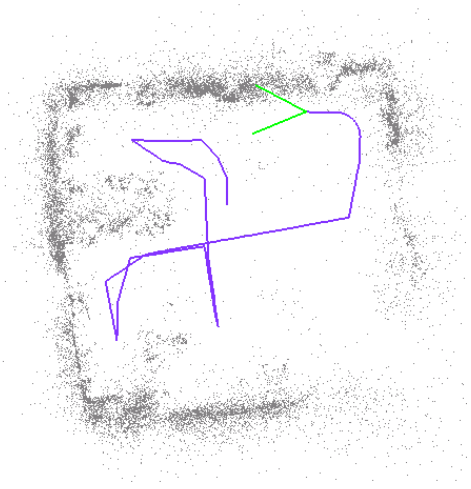


Fig. 1. Bird's eye view of $3D$ landmark map used for localization. Only the $3D$ coordinates of each landmark are shown as dark dots. Overlayed is the path the robot followed during map construction while the robot's position at the end of the path is shown using a green "V".

In this case, the importance factors are given by [13]

$$w^{(i)} = \pi(y_t)\pi(\tilde{s}_t^{(i)}|u_t) Bel(s_{t-1}^{(i)}) \quad (3)$$

MCL and dual MCL are complementary and it is shown in [13] that using a mixture of the two results in superior performance compared to either of them. Implementing a particle filter is straightforward. One must describe the initial belief, the proposal distribution and the observation and motion models. Until now, MCL methods have been limited to solving the simpler case of planar robots with only 3 dof. In the next few sections we will provide our solution, σMCL, for vision-based robots moving unconstrained in $3D$ space with 6 dof. However, first we will describe the maps we use and how we construct them since they play a central role in our approach.

III. Map Construction

We use maps of naturally occurring $3D$ landmarks as proposed in [14]. Each landmark is a vector $l = \{P, C, \alpha, s, f\}$ such that $P = \{X^G, Y^G, Z^G\}$ is a 3-dimensional position vector in the map's global coordinate frame, C is the 3×3 covariance matrix for P, and α, s, f describe an invariant feature based on the Scale Invariant Feature Transform (SIFT) [3]. Parameter α is the orientation of the feature, s is its scale and f is the 128-dimensional key vector. SIFT descriptors have been shown [15] to outperform others in matching accuracy and as such they are a natural choice for this application. An example of a map with $32,000$ landmarks is shown in Figure 1. We constructed it using a total of 1200 frames while the robot traveled a total distance of 25 meters.

We learn these maps using the method presented in [14]. We use visual measurements to correct the odometry estimate as the robot travels. We assume that each landmark is independent and its position is tracked using a Kalman filter.

This approach is only useful for constructing maps for small environments [14].

IV. Observation Function

In order to implement the Monte-Carlo algorithm we must first specify the distribution $p(y_t|s_t)$ that is used to compute the importance weights. Given our map representation and our visual sensor, a measurement y_t consists of correspondences between landmarks in the current view and landmarks in the known map.

Let I_t^R and I_t^L denote the right and left gray scale images captured using the stereo camera at time t. The right camera is the reference camera. We compute image points of interest from both images by selecting maximal points in the scale space pyramid of a Difference of Gaussians [3]. For each such point, we compute the SIFT descriptor and record its scale and orientation. We then match the points in the left and right images in order to compute the points' $3D$ positions in the camera coordinate frame. Matching is constrained by the stereo camera's known epipolar geometry and the Euclidean distance of their SIFT keys. Thus, we obtain a set $O_C = \{o_1, o_2, \cdots, o_n\}$ of n local landmarks such that $o_j = \{P_{o_j} = \{X_{o_j}^L, Y_{o_j}^L, Z_{o_j}^L\}, p_{o_j} = \{r_{o_j}, c_{o_j}, 1\}, C, \alpha, s, f\}$ where $p_{o_j} = \{r_{o_j}, c_{o_j}, 1\}$ is the image coordinates of the point and $j \in [1 \cdots n]$.

An observation is defined as a set of k correspondences between landmarks in the map and the current view, $y_t = \cup_{1 \cdots k}\{l_i \leftrightarrow o_j\}$ such that $i \in [1..m]$ and $j \in [1..n]$ where m is the number of landmarks in the map and n is the number of landmarks in the current view. We compare the landmarks' SIFT keys in order to obtain these correspondences just as we did before during stereo matching. There are no guarantees that all correspondences are correct but the high specificity of SIFT results in a reduced number of incorrect matches.

A pose of the camera, s_t, defines a transformation $[R, T]_{s_t}$ from the camera to the global coordinate frame. Specifically, R is a 3×3 rotation matrix and T is a 3×1 translation vector. Each landmark in the current view can be transformed to global coordinates using the well known equation

$$P_{o_j}^G = R_{s_t} P_{o_j} + T_{s_t} \tag{4}$$

Using equation 4 and the Mahalanobis distance metric (in order to take into account the map's uncertainty), we can define the observation density by:

$$p(y_t|s_t) = e^{0.5 \sum_{b=1}^{k} (P_{o_j}^{G,b} - P_i^{G,b})^T C^{-1} (P_{o_j}^{G,b} - P_i^{G,b})} \tag{5}$$

where C is given by:

$$C = R_{s_t} C_{o_j} R_{s_t}^T + C_i \tag{6}$$

V. Computing Camera Motion

Another essential component to the implementation of MCL is the specification of the robot's motion model, u_t. In all previous work, this has been a function of the robot's odometry, i.e., wheel encoders that measure the amount the robot's wheels rotate that can be mapped to a metric value of displacement and rotation. Noise drawn from a Gaussian is then added to this measurement to take into account slippage as the wheels rotate. Such measurements although accurate and available on all research robots are only useful for planar motions. We want to establish a more general solution. Thus, we obtain u_t measurements by taking advantage of the vast amount of research in multiple view geometry. Specifically, it is possible to compute the robot's displacement directly from the available image data including an estimate of the uncertainty in that measurement.

Let I_t and I_{t-1} represent the pairs of stereo images taken with the robot's camera at two consecutive intervals with the robot moving between the two. For each pair of images we detect points of interest, compute SIFT descriptors for them and perform stereo matching, as described earlier in section IV, resulting in 2 sets of landmarks L_{t-1} and L_t. We compute the camera motion in two steps, first by the linear estimation of the Essential matrix, E, and its parameterization, as described below. Second, we compute a more accurate estimate by dropping the linearity assumption and using a non-linear optimization algorithm minimizing the re-projection error of the $3D$ coordinates of the landmarks.

A. Linear Estimation of the Essential Matrix

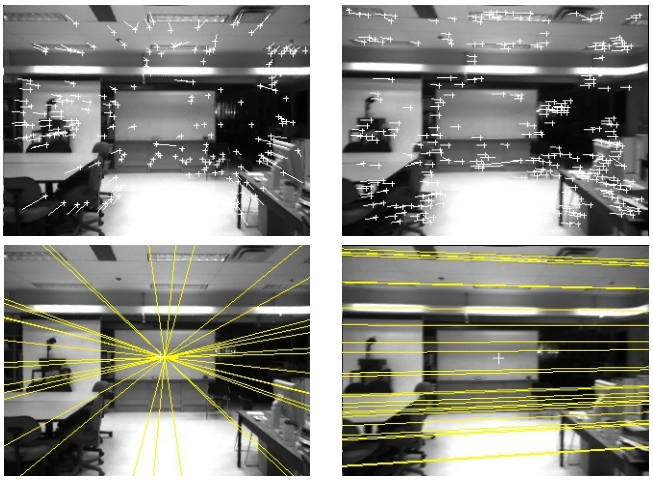
For this part we only consider the images from the reference cameras for times t and $t-1$. Using the SIFT descriptors we obtain landmark correspondences between the two images as described earlier in section IV. Let the i-th such pair of landmarks be denoted $l_t^i \leftrightarrow l_{t-1}^i$. For the time being we only consider the image coordinates of these landmarks, $p_t^i \leftrightarrow p_{t-1}^i$.

Given this set of $2D$ point correspondences, the *Essential* matrix, E, is any 3×3 matrix that satisfies the property

$$p_t^{i\,T} E p_{t-1}^i = 0 \tag{7}$$

The Essential matrix maps points from one image to lines on the other. If it is known, the camera pose, $[R, T]$, at time t can be estimated with respect to the camera pose at time $t-1$ via its parameterization. We use the *normalized 8-point* algorithm to estimate E. The algorithm can be found in most modern machine vision books such as [16], and so we do not repeat here. We note that the algorithm requires a minimum of 8 point correspondences. In our experiments, we obtain an average of 100 such correspondences allowing us to implement the robust version of the algorithm that uses RANSAC to consider solutions for subsets of them until a solution is found with a high number of inliers.

We can compute the camera pose via the Singular Value Decomposition (SVD) of E as described in [16]. As a result, we obtain the rotation matrix $\tilde{R}$ and a unit size vector $\tilde{T}$ that denotes the camera's displacement direction but not the actual camera displacement. Although we could take advantage of the information in $\tilde{T}$ to guide our non-linear solution described in the next section, currently we do not and we simply set $\tilde{T} = 0$. Figure 2 gives two examples of the estimated epipolar geometry for forward motion and rotation.



(a) (b)

Fig. 2. Examples of estimating the epipolar geometry for (a) forward motion and (b) a rotation to the right. The top row shows the point correspondences between two consecutive image frames. Crosses mark the points at time $t-1$ and lines point to their location at time t. The bottom row, shows the epipolar lines drawn for a subset of the matched points using the estimated Essential matrix.

We use $[\tilde{R}, \tilde{T}]_{s_t}$ to initialize the non-linear estimation of the camera pose. The advantage of the initial value is that it allows us to do outlier removal , i.e., we can remove landmark matches that do not satisfy equation 7. Additionally, having an initial estimate helps guide the non-linear optimization algorithm away from local minima.

B. Non-Linear Estimation of the Camera Pose

Given an initial value for the camera pose at time t with respect to the camera at time $t-1$, we compute a more accurate estimate using the Levenberg-Marquardt (LM) non-linear optimization algorithm. We utilize the $3D$ coordinates of our landmarks and use the LM algorithm to minimize their re-projection error. Let $\tilde{x}_t$ be the 6-dimensional vector $x_t = [roll, pitch, yaw, T_{11}, T_{21}, T_{31}]$ corresponding to a given $[R, T]$. Our goal is to iteratively compute a correction term χ

$$x_t^{i+1} = x_t^i - \chi \tag{8}$$

such as to minimize the vector of error measurement ϵ , i.e., the re-projection error of our $3D$ points. For a known camera calibration matrix K, ϵ is defined as

$$\epsilon = \begin{bmatrix} \epsilon_0^T \\ \epsilon_1^T \\ \vdots \\ \epsilon_k^T \end{bmatrix} = \begin{bmatrix} p_t^0 - K(RP_{t-1}^0 + T) \\ p_t^1 - K(RP_{t-1}^1 + T) \\ \vdots \\ p_t^k - K(RP_{t-1}^k + T) \end{bmatrix} \tag{9}$$

Given an initial estimate for the parameters, we wish to solve for χ that minimizes ϵ , i.e.,

$$\begin{bmatrix} J \\ \lambda I \end{bmatrix} \chi = \begin{bmatrix} \epsilon \\ \lambda d \end{bmatrix} \Leftrightarrow (J^T J + \lambda I)\chi = J^T \epsilon + \lambda I d \tag{10}$$

where $J = [\frac{\partial \epsilon_0}{\partial \chi}, \cdots, \frac{\partial \epsilon_k}{\partial \chi}]^T$, is the Jacobian matrix,I is the identity matrix and d is the initial solution from the linear estimate given by $[\tilde{R}, \tilde{T}]$. The LM algorithm introduces the variable λ that controls the convergence of the solution by switching between pure gradient descent and Newton's method. As discussed in [17] solving 10 , i.e., the normal equations, minimizes

$$||J\chi - \epsilon||^2 + \lambda^2 ||\chi - d||^2 \tag{11}$$

The normal equations can be solved efficiently using the SVD algorithm. A byproduct from solving 11 is that we also get the covariance of the solution in the inverse of $J^T J$. Table I compares our vision-based estimate of camera motion with that of odometry. One can see that it is very accurate.

Odometry			Vision-based Estimate					
x	y	θ	x	y	z	α	θ	β
0.00	0.00	-1.76	0.01	0.00	-0.01	0.22	-1.83	0.05
0.00	0.00	-2.11	-0.01	-0.00	0.01	-0.01	-1.97	-0.02
0.00	0.00	2.11	0.01	0.02	0.00	0.01	1.63	0.00
0.12	0.00	0.00	0.18	-0.02	0.01	-0.01	-0.13	0.18
0.11	0.01	0.00	0.09	0.01	0.01	-0.26	-0.52	0.04

TABLE I

COMPARING OUR LEAST-SQUARES ESTIMATE OF CAMERA MOTION WITH ODOMETRY. POSITION IS SHOWN IN cm AND ORIENTATION IS DEGREES.

VI. THE MIXTURE PROPOSAL DISTRIBUTION

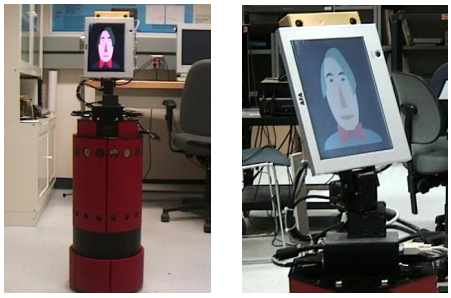
As we discussed in section II, we perform filtering using a mixture of the MCL and dual MCL proposal distributions [1]

$$(1-\phi)\tilde{q}_t + \phi q_t = (1-\phi)p(y_t|\tilde{s}_t) + \phi p(s_t|s_{t-1}, u_t) \tag{12}$$

where ϕ is known as the mixing ratio and we have set it to 0.80 for all of our experiments.

Sampling from $p(s_t|s_{t-1}, u_t)$ is straightforward as all particles from time $t-1$ are updated using our estimate of the camera's motion, u_t, with noise added drawn from our confidence on this estimate given by $(J^T J)^{-1}$. On the other hand, sampling from the dual proposal distribution is thought of as a hard problem in robotics since we must be able to generate poses from observations. It turns out that for our choice of maps and sensor this is trivial.

Let M be a map of m landmarks l_i^G for $1 \leq i \leq m$ and N be the set of n landmarks l_j^L for $1 \leq j \leq n$ in the current view. Let y_t be the current observation. Using the procedure described in section V and k random subsets of matched landmarks, $x_{1:k} \subset X$, we compute k candidate poses $\tilde{s}_t^{(1:k)}$. For efficiency we only call on the non-linear estimation procedure initializing the pose to zero translation and rotation. Even if our subset of landmarks leads to an incorrect estimate, this sample will receive a low weight given our complete set of observations and as such we do not need to incur the penalty of the robust linear estimate described in section V-A. For each sampled pose we compute $p(y_t|\tilde{s}_t^{(j)})$ with $1 \leq j \leq k$ using equation 5. Because of the high quality of observations, we only need to sample as few as 100 poses to get a good approximation for $p(y_t|\tilde{s}_t)$. To sample from the dual proposal we draw a random particle from $\{\tilde{s}_t^{(1:k)}, \tilde{w}_t^{1:k}\}$ according to



(a) (b)

Fig. 3. The robot we used for our experiments (a) seen from a distance and (b) closeup of its head

the particle weights and compute its importance factor using equation 3.

Lastly we should mention that the procedure we just described for approximating the dual proposal distribution is also useful for estimating $p(s_0)$, that is, the initial belief. It is common practice that the initial belief is set to be a uniform distribution but considering the large dimensionality of our state space, ours is a better choice.

VII. Single-frame Approach using RANSAC

A straightforward approach to global localization that does not require filtering is presented in [18]. Poses are generated as discussed in the previous section and evaluated using equation 5. For robustness a RANSAC approach is employed in that poses are sampled until one is found that is well supported by the current observation, i.e., $p(y_t|s_t)$ is above a given threshold.

The advantage of such an approach, made possible by the geometry of the camera and map representation, is its simplicity and its high accuracy considering that only a single image frame needs to be considered. A major disadvantage is that is not suitable for tracking the robot's pose over multiple frames due to the large variance of the estimates. Additionally, a user must specify beforehand the number of samples to consider ([18] uses only 10) and a threshold value for what is a good pose. However, the former can be avoided if the adaptive RANSAC algorithm is used.

Until now, no vision-based MCL method has been shown to match the accuracy of the single-frame approach. We will show in the next section that σMCL comes close to this target while considering a more general case of unconstrained motion in $3D$.

VIII. Experiments

We have implemented *σMCL* for our mobile robot seen in Figure 3. It is a modified RWI B-14 quasi-holonomic base. It is equipped with a PointGrey Research Bumblebee stereo vision camera mounted on a Pan-Tilt Unit. It has an on-board Pentium-based computer and wireless connectivity. It has no other functioning sensors other than the stereo camera.

For our experiments, we have used 3 different sets of images, S_1, S_2, S_3, at 320×240 resolution. S_1 is the set of images that we used for constructing the map. It is a useful set because along with the images we have the robot's pose at the time they were acquired. We use these corrected odometry estimates as a baseline for judging the accuracy of the *σMCL* approach. One obvious disadvantage of using this set is that we always get a really large and accurate number of landmark correspondences. In order to show that *σMCL* works in general, the second set of images, S_2, is an arbitrary sequence that was not used during map building. Unfortunately, both of these image sets are acquired as the robot traverses a planar path. To demonstrate that our solution works for non-planar motions, we have acquired a $3rd$ set of images, S_3, in which we moved the camera by hand in a rectangular pattern off the robot's plane of motion.

An important and open question in the study of particle filters is the number of particles to use. In our experiments with S_1 we found that as few as 100 particles were sufficient to achieve highly accurate results; we can get away with only a small number of particles because of good observations. Using more particles provided no improvement. For our experiments with S_2 and S_3 we used 500 particles; using more did not generate a noticeable improvement on the computed trajectory of the robot. These numbers are significantly smaller than other vision-based approaches. For example, [6] uses 5000 particles.

Figure 4 shows the results of global localization using S_1. Part (a) of the figure presents the initial belief. Part(b) shows the particles after 7 frames, the robot having moved forward for about $80cm$. One can see that there are several modes to the distribution. Part (c) shows the samples after 35 frames having all converged to the robot's true location. The robot has moved a total of $100cm$ and rotated by 20 degrees.

Figure 5 plots the localization error for the mean pose taken over all the particles with respect to the odometry estimate. The position error is less than $20cm$ and the orientation error is less than 6 degrees. The results are the average of 10 runs per frame. In comparison, [13] reports a localization error that varies from $100cm$ to $500cm$. Similarly, the results presented in [6] specify an error that is as large as $82cm$ in position and 17 degrees in orientation. In [9] the position error varies from $45cm$ to $71cm$. Finally, [8] reports that the localization error is $20cm$ but this value is highly correlated with the distance of the reference images which is also $20cm$. [13], [9], [8] do not include results for the error with respect to the robot's orientation. Also shown in Figure 5 is the estimate of the single frame approach using adaptive RANSAC as described in section VII. It performs better than σMCL but the generated track is more noisy due to the large variance of the estimates.

Figure 6 shows the camera path for the image set S_2. It was generated using a total of 500 images while the robot traversed a distance of 18 meters. It took about 20 frames before the robot estimated its location and it successfully kept track of it from then on.

Finally, Figure 7 shows the results using S_3. We have provided a general solution to the localization problem that can handle unconstrained motions in $3D$. So far we have only

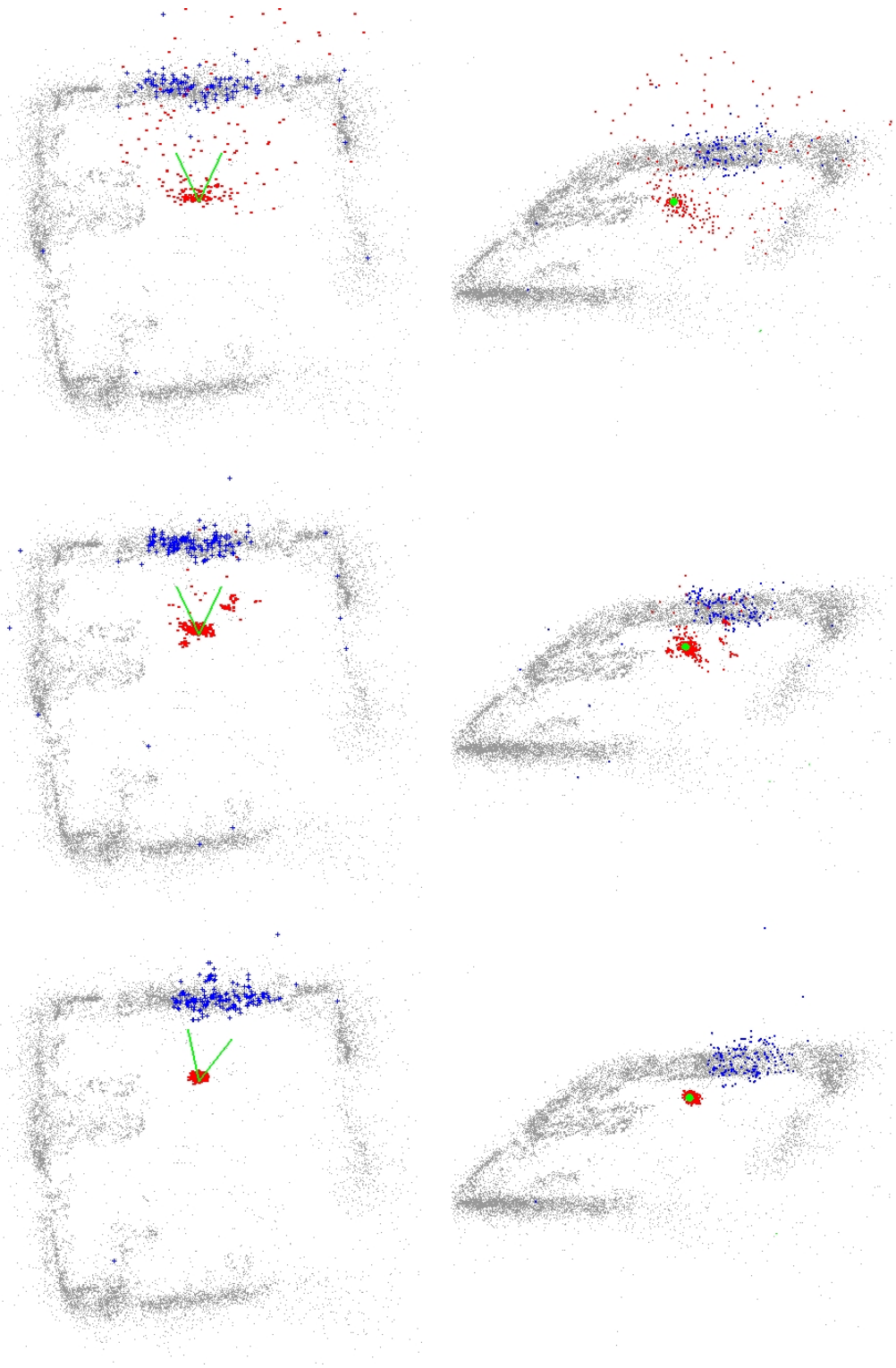


Fig. 4. An example of the evolution of particles during global localization. The left column shows the state from a top view point and the right column shows it in $3D$. The current observation is marked using blue crosses/dots. The particles are shown in red and the robot's true position is shown using a green "V" on the top row and a green sphere on the bottom. Row (a) shows $p(s_0)$, (b) shows the sample distribution after 7 frames and (c) shows it after 35 frames.

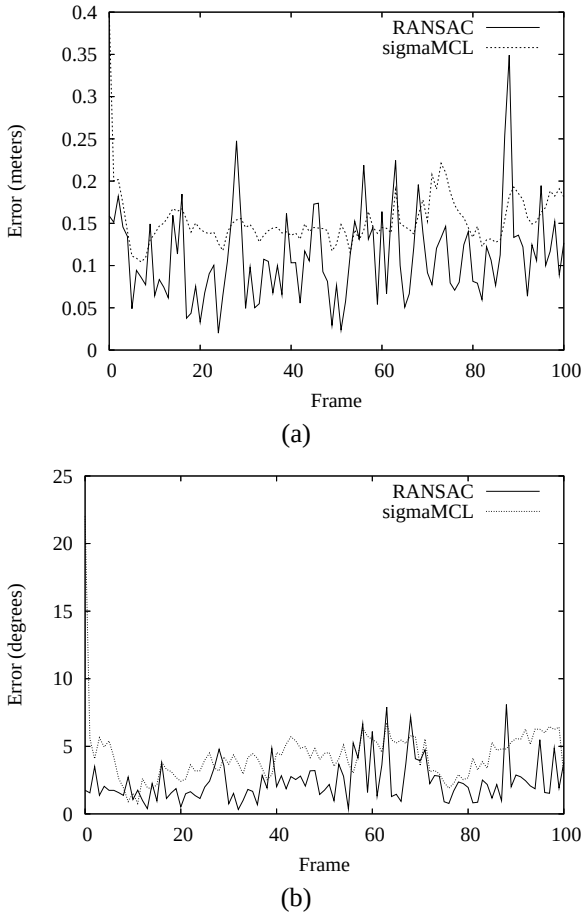


Fig. 5. Plots of the localization error for *σMCL* and single-frame adaptive RANSAC approach. Part (a) shows the mean error in position and (b) shows the mean error in orientation for 100 frames. Both approaches are equally accurate but the standard deviation of the *σMCL* is less than $1/3$ of RANSAC.

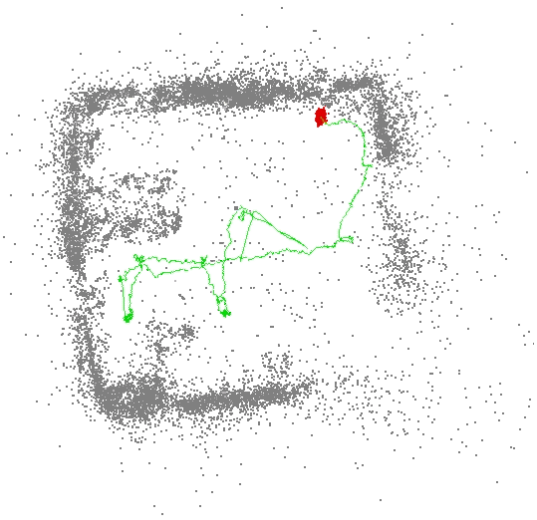


Fig. 6. The robot's path for image set S_2. The robot traveled a total distance of 18 meters.

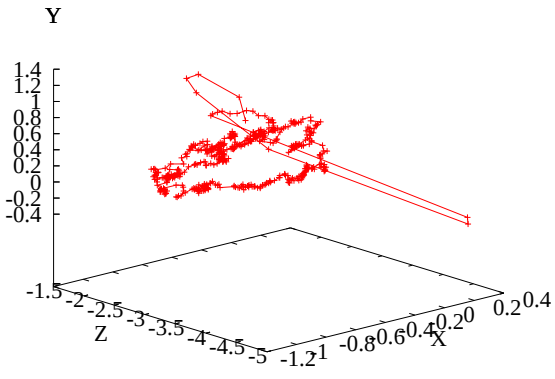


Fig. 7. The robot's path for image set S_3. The camera's path is correctly tracked around an approximately rectangular pattern on the $x-y$ plane, after the first few frames required until σMCL converges.

demonstrated it using data from our robot that moves on a planar surface and has limited range off this plane due to its steerable head. However, in order to demonstrate that our approach really works we present results with an image set obtained by moving the camera around by holding it in our hand. This particular image set is very challenging with much jitter in the camera motion. The robot lives on the $x-z$ plane and we obtained S_3 by moving the camera on the $x-y$ plane in an approximately rectangular pattern. In Figure 7, the first few frames show a large variation in the $x-z$ plane until σMCL converges, after which the camera's path is correctly tracked along the rectangular path.

We timed the performance of our approach on a $3.2GHz$ Pentium 4. σMCL takes on average $1.1secs$ per frame. The time does not include computing landmark correspondences, i.e., y_t, because it is common to both approaches. In our implementation, it takes about $1sec$ to compute y_t because we have not implemented the efficient kd-tree approach for SIFT matching as in [3]. The performance of σMCL is good enough for an online system even though we have not made much effort to optimize our software.

For completeness, Figure 8 shows the results for the *kidnapped robot* problem using S_1. We demonstrate that the robot can quickly re-localize after a sudden loss in position and a strong prior estimate of its location. Specifically, we allowed the robot to localize itself and then we transported it at frame 30. The robot localized again within 20 frames. We repeated this procedure at frame 80 and the robot once again estimated its position within the error bounds reported earlier after only 30 frames.

IX. Conclusion

We have presented an approach to vision-based Monte-Carlo localization with a mixture proposal distribution that uses the best of invariant image-based landmarks and statistical techniques. Our approach decouples the sensor from the

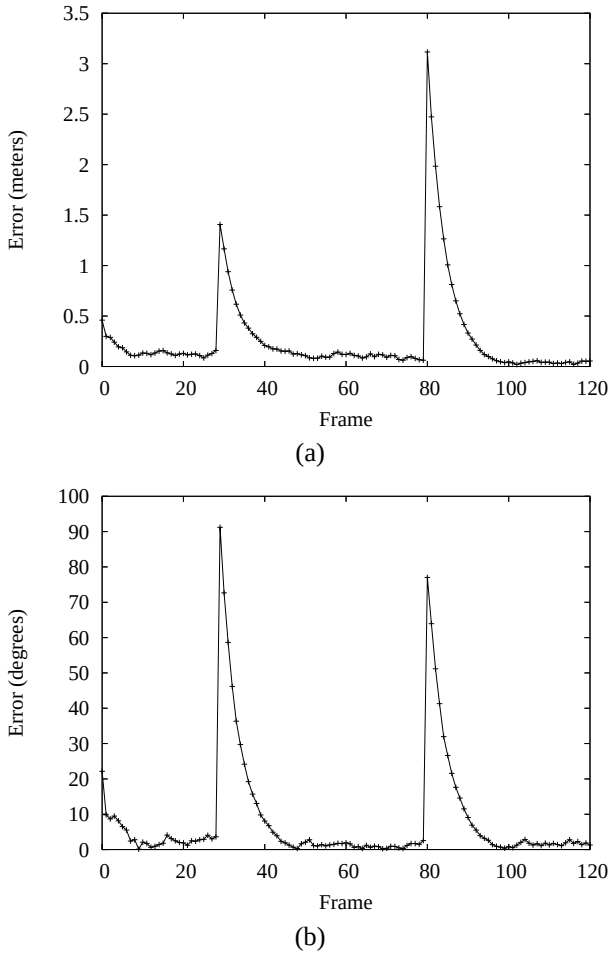


Fig. 8. Results for the kidnapped robot problem. We *kidnapped* the robot twice at frames 30 and 80. Part (a) shows the error for the robot's position and (b) shows it for the orientation.

robot's body by specifying the motion model as a function of sensor measurements and not robot odometry. As a result we can solve for unconstrained 6 dof motion by taking advantage of results in multiple view geometry. We presented a number of experimental results to prove that our approach is robust, accurate and efficient.

In the future, we plan to optimize the efficiency of our implementation using such improvements as the kd-tree approach for landmark matching [3]. For all our experiments we used a fixed value for the mixing ratio. It would be worth experimenting with adapting this ratio according to the variance of the two proposal distributions. Finally, we would like to apply our approach towards the *simultaneous localization and mapping* (SLAM) problem possibly through the implementation of the FastSLAM [19] algorithm.

ACKNOWLEDGMENT

The authors would like to thank Nando de Freitas, Kevin Murphy, Jesse Hoey and Rob Sim for their helpful comments on an earlier draft of this paper.

REFERENCES

[1] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte-Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.

[2] L. Chrisman, "Reinforcement learning with perceptual aliasing: The perceptual distinctions approach," in *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*. San Jose, California: AAAI Press, 1992, pp. 183–188.

[3] D. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, Kerkyra, Greece, September 1999, pp. 1150–1157.

[4] H. Moravec and A. Elfes, "High-resolution maps from wide-angle sonar," in *Proc. IEEE Int'l Conf. on Robotics and Automation*, St. Louis, Missouri, Mar. 1985.

[5] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'99)*, Fort Collins, CO, June 1999.

[6] J. Wolf, W. Burgard, and H. Burkhardt, "Robust vision-based localization for mobile robots using an image retrieval system based on invariant features," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[7] R. Sim and G. Dudek, "Comparing image-based localization methods," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI '03)*, Acapulco, Mexico, 2003.

[8] E. P. E. Menegatti, M. Zoccarato and H. Ishiguro, "Image-based Monte-Carlo localisation with omnidirectional images," *Robotics and Autonomous Systems*, vol. 48, no. 1, pp. 17–30, August 2004.

[9] H. Gross, A. Koenig, C. Schroeter, and H. Boehme, "Omnivision-based probalistic self-localization for a mobile shopping assistant continued," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, Las Vegas, USA, 2003, pp. 1505–1511.

[10] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proc. of the IEEE International Conference on Robotics & Automation (ICRA '02)*, San Francisco, CA, April 2000, pp. 1023–1029.

[11] B. Krose, R. Bunschoten, S. Hagen, B. Terwun, and N. Vlassis, "Household robots look and learn," *IEEE Robotics and Automation Magazine*, vol. 11, no. 4, pp. 45–52, December 2004.

[12] T. Rofer and M. Jungel, "Vision-based fast and reactive Monte-Carlo localization," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'03)*, Tapei, Tawiwan, 2003, pp. 856–861.

[13] S. Thrun and D. Fox, "Monte-Carlo localization with mixture proposal distribution," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Austin, TX: AAAI, 2000.

[14] S. Se, D. Lowe, and J. J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant landmarks," *International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, August 2002.

[15] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *International Conference on Computer Vision & Pattern Recognition*, vol. 2, June 2003, pp. 257–263. [Online]. Available: http://lear.inrialpes.fr/pubs/2003/MS03

[16] R. Hartley and A. Zisserman, *Multiple View Geometry, in computer vision*. Cambridge University Press, 2000.

[17] D. Lowe, "Fitting parameterized three-dimensional models to images," *IEEE Trans. Pattern Analysis Mach. Intell. (PAMI)*, vol. 13, no. 5, pp. 441–450, May 1991.

[18] S. Se, D. Lowe, and J. Little, "Local and global localization for mobile robots using visual landmarks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '01)*, Maui, Hawaii, October 2001.

[19] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.