$\sigma SLAM$: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution

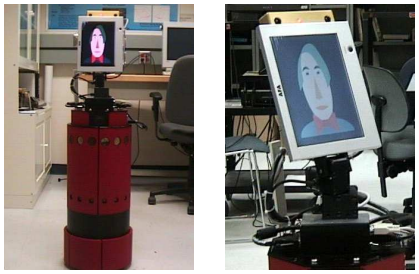
Pantelis Elinas, Robert Sim, James J. Little
Computer Science Department, University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
Email: {elinas, simra, little}cs.ubc.ca

***Abstract*— We consider the problem of Simultaneous Localization and Mapping (SLAM) using the Rao-Blackwellised Particle Filter (RBPF) for the class of indoor mobile robots equipped only with stereo vision. Our goal is to construct dense metric maps of natural 3D point landmarks for large cyclic environments in the absence of accurate landmark position measurements and motion estimates. Our work differs from other approaches because landmark estimates are derived from stereo vision and motion estimates are based on sparse optical flow. We distinguish between landmarks using the Scale Invariant Feature Transform (SIFT). This is in contrast to current popular approaches that rely on reliable motion models derived from odometric hardware and accurate landmark measurements obtained with laser sensors. Since our approach depends on a particle filter whose main component is the proposal distribution, we develop and evaluate a novel mixture proposal distribution that allows us to robustly close large loops. We validate our approach experimentally for long camera trajectories processing thousands of images at reasonable frame rates.**

I. Introduction

In robotics, the problem of *Simultaneous Localization and Mapping* (SLAM) is that of estimating both a robot's location and a map of its surrounding environment. It is an inherently hard problem because noise in the estimate of the robot's pose leads to noise in the estimate of the map and vice versa. In general, SLAM algorithms must address issues regarding sensors, map representation and robot/environment dynamics. A probabilistic framework is necessary for combining over time the incoming sensor measurements and robot control signals. There are two such frameworks popular in the SLAM community, the Extended Kalman Filter and the Rao-Blackwellised Particle Filter. We choose to study the latter since it scales better for maps with large numbers of landmarks and it handles incorrect data associations more gracefully.

In this paper, we present $\sigma SLAM$ (where σ stands for stereo), a novel model of the RBPF applied to the SLAM problem for robots with stereo vision. Our work is unique because it uses a video stream of stereo images as the **only input** while it achieves results whose accuracy is comparable to those of other models that rely on laser and odometric measurements. We derive an estimate of the robot's motion from sparse visual measurements using stereo vision and multiple view geometry techniques. We construct dense metric maps of $3D$ point landmarks identified using SIFT [1] whose invariance to changes in image translation, scaling and rotation makes it suitable for reliable, but not perfect, data association. SIFT's invariance properties along with our novel mixture proposal distribution improve the robustness of our filter in closing large loops. We show that our model works well in environments with large changes in illumination, image blurring, glass doors and many large windows. Finally, we show that we can trivially compute $2D$ occupancy grids, if desired. Such grids are useful for robot path planning and obstacle avoidance.



(a) (b)

Fig. 1. The RWI B-14 robot that we used for data collection seen from (a) far and (b) closeup view of the stereo camera.

In comparison, previous work on SLAM has focused predominantly on robots that utilize $2D$ occupancy grid maps [2] and use laser sensors [3], [4]. Laser has high depth resolution providing accurate measurements of landmark positions but its low perceptual resolution makes it difficult to distinguish between different landmarks. We focus on using vision as the sensing modality. Vision can be used to construct $2D$ occupancy grids [5] and as such one might believe that the application of laser-based SLAM algorithms could be used just as well with stereo vision. However, as shown in part (b) of Figure 4 its low depth resolution makes it very difficult to directly apply such occupancy grid methods to vision-based SLAM. Instead, we focus on learning maps of $3D$ natural landmarks identifiable by their appearance in images. We demonstrate how we can trivially construct the $2D$ occupancy grids as a by-product of our algorithm.

Our work is analogous to that of [6] because we learn a similar type of map. However, their approach maintains only the maximum likelihood estimate of the camera pose predicted by robot odometry and corrected using visual information, that is $3D$ points identified using SIFT. They exploit odometry information to achieve better data associations. This approach has only been shown to work for small environments and trajectories. A similar method that constructs metric maps is presented in [7]. Their algorithm is a direct extension of FastSLAM [4] for visually guided robots that identifies landmarks using SIFT. They construct maps similar to those in [6]. They present results that are promising but preliminary and only for 1 particle in which case their approach closely resembles the original work of [6]. Some recent work is also focused on the use of monocular cameras. For example, [8] uses a particle filter with a single camera and robot odometry to map small indoor environments. They construct topological maps and identify visual landmarks using SIFT. In this case, occupancy grid maps would have to be constructed using a different sensor, i.e., laser, in which case using vision for SLAM seems unnecessary since this can be achieved using the laser data in the first place. Work on monocular SLAM that does not depend on robot odometry has been presented in [9]. Their approach is mostly tailored to real-time performance. In consequence, they identify visual landmarks using salient visual features that are not necessarily distinctive enough for operation in large environments. This approach is also based on the Extended Kalman Filter that is inefficient because the filter scales quadratically in the number of landmarks.

This paper is structured as follows. We first present an overview of Bayesian filtering applied to SLAM and its RBPF approximation. We then focus on the details of our vision-based SLAM presenting our map representation, observation and motion models. We discuss our novel mixture proposal distribution and give the complete algorithm for our particle filter. We then provide experimental results that support our hypothesis in terms of robot localization accuracy and our ability to construct occupancy grid maps for navigation. We conclude by discussing directions for future work.

II. SLAM USING THE RAO-BLACKWELLLIZED PARTICLE FILTER

Formally, let s_t denote the robot's pose at time t, m_t the map learned thus far and $x_t = \{s_t, m_t\}$ be the complete *state*. Also, let u_t denote a control signal or a measurement of the robot's motion from time $t-1$ to time t and z_t be the current observation. The set of observations and controls from time 0 to t are denoted by z^t and u^t respectively. Our goal is to estimate the density

$$p(s_t, m_t|z^t, u^t) = p(x_t|z^t, u^t) \tag{1}$$

That is, we must integrate the set of observations and controls as they arrive over time in order to compute the posterior probability over the unknown *state*. Applying Bayes rule on Equation 1 and making the Markov assumption we get [10]

$$p(x_t|z^t, u^t) = Bel(x_t) =$$
$$\eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) p(x_{t-1}|z^{t-1}, u^{t-1}) dx_{t-1} =$$
$$\eta p(z_t|x_t) \int p(x_t|u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1} \tag{2}$$

where η is a normalizing constant.

Equation 2 allows us to recursively estimate the posterior probability of maps and robot poses if the two distributions $p(z_t|x_t)$ and $p(x_t|u_t, x_{t-1})$ are given. These distributions model the observations and the state's evolution respectively. For SLAM, an analytical form for $Bel(x_t)$ is hard to obtain and as such the Bayes filter is not directly applicable. Instead, it was noted in [4], [11] that if we estimate the modified posterior, $p(x^t|z^t, u^t)$, then we don't have to integrate and we can obtain a solution efficiently using a Particle Filter.

Particle Filtering is a general method for approximating $p(x^t|z^t, u^t)$ using a set of m weighted particles, $\{x^{(i)}, w^{(i)}\}_{i=1,\cdots,m}$. The system is initialized according to $p(x_0)$ and the recursive update of the Bayes filter proceeds using a procedure known as *sampling-importance-resampling* [12].

A major caveat of the standard PF is that it requires a very large number of particles as the size of the *state* increases. Since for SLAM the *state* of the system includes the map that often has tens of thousands of landmarks, the PF is not applicable from a practical point of view. The method of Rao-Blackwellization can be used to reduce the complexity of the PF by factoring the posterior [11], [13]

$$p(s^t, m_t|z^t, u^t) = p(s^t|z^t, u^t) \prod_k p(m_t(k)|s^t, z^t, u^t) \tag{3}$$

where $m_t(k)$ denotes the k-th landmark in the map. We use a standard PF to estimate $p(s^t|z^t, u^t)$ and an EKF for each of the k landmarks.

III. VISION-BASED SLAM

In this section we present our model for vision-based SLAM. We first describe how we represent maps that are central to our method. Next we define observations and how we compute the observation likelihood, followed with a description of our motion model based on visual odometry. We then discuss our mixture proposal distribution and give the complete particle filter algorithm.

A. Map Representation

We construct maps of naturally occurring $3D$ landmarks similar to those proposed in [6]. Each landmark is a vector $l = \{P^G, C^G, \alpha, s, f\}$ such that $P^G = \{X^G, Y^G, Z^G\}$ is a 3-dimensional position vector in the map's global coordinate frame, C^G is the 3×3 covariance matrix for P^G, and α, s, f describe an invariant feature based on the Scale Invariant Feature Transform [1]. Parameter α is the orientation of the feature, s is its scale and f is the 128-dimensional key vector which represents the histogram of local edge orientations.

B. Observation Model

Let I_t^R and I_t^L denote the right and left gray scale images captured using the stereo camera at time t. We compute image points of interest from both images by selecting maximal points in the scale space pyramid of a Difference of Gaussians [1]. 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 $o_j = \{P_{o_j}^L = \{X_{o_j}^L, Y_{o_j}^L, Z_{o_j}^L\}, p_{o_j}^L = \{r_{o_j}, c_{o_j}, 1\}, C^L, \alpha, s, f\}$ where $p_{o_j}^L = \{r_{o_j}, c_{o_j}, 1\}$ 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, $z_t = \cup_{1 \cdots k}\{l_i \leftrightarrow o_j\}$ 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. Each local landmark either corresponds to a mapped landmark l_k, or has no corresponding landmark, denoted by the null correspondence $l_\emptyset$. 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

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

Using Equation 4 and the Mahalanobis distance metric we can define the observation log-likelihood, $\log p(z_t|m_t, s_t)$. Special consideration must be taken when computing this quantity, particularly where large numbers of feature observations, with significant potential for outlier correspondences, are present. We compute it by summing over the feature correspondences:

$$\log p(z_t|m_t, s_t) = \sum_k \log p(o_k|l_k, s_t) \tag{5}$$

The log-likelihood of the k-th observation is given by

$$\log p(o_k|l_k, s_t) = -0.5 \min(T_l, (P_{o_k}^G - P_k^G)^T S^{-1} (P_{o_k}^G - P_k^G)) \tag{6}$$

where the correspondence covariance S is given by the sum of the transformed local landmark covariance $C_{o_k}^L$ and the map landmark covariance C_k^G:

$$S = R_{s_t} C_{o_k}^L R_{s_t}^T + C_k^G. \tag{7}$$

For the null correspondence, S is assumed to be zero.

The maximum observation innovation T_l is selected so as to prevent outlier observations from significantly affecting the observation likelihood. However, given the potentially large numbers of correspondences, even with a reasonable setting for T_l (in our case, 4.0), the magnitude of $\log p(z_t|m_t, s_t)$ can be such that raising it to the exponential results in zero values. Since particle weights are computed partially based on the observation likelihood, it is important that the computation of this quantity is robust to outliers. We have shown in previous work [14] that this can be done correctly and without loss of generality by subtracting the observation likelihood of the least likely particle. Similarly, in the same work [14], we provide details as to how a landmark is evaluated as stable enough to be added to the map.

C. Motion Model

An essential component to the implementation of RPBF is the specification of the robot's motion model, u_t. In the vast majority of previous SLAM 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. Odometric measurements of this type are limited to robots moving on planar surfaces. 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 [15]. 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. This is often known as visual odometry [16], [17], [18], [19], [20].

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 III-B, resulting in 2 sets of landmarks L_{t-1} and L_t. We compute the camera motion using the Levenberg-Marquardt (LM) algorithm minimizing the re-projection error of the $3D$ coordinates of the landmarks [15]. Let $\tilde{s}_t$ be the 6-dimensional vector $\tilde{s}_t = [roll, pitch, yaw, T_{11}, T_{21}, T_{31}]$ corresponding to a given $[R, T]$. Our goal is to iteratively compute a correction term χ

$$\tilde{s}_t^{i+1} = \tilde{s}_t^i - \chi \tag{8}$$

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.,

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

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 an initial solution that in this case is set to zero rotation and translation. 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 [21] solving Equation 10 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 Equation 11 is that we also get the covariance of the solution in the inverse of $J^T J$.

IV. The Proposal Distribution

The correct performance of a particle filter highly depends on the selection of proposal distribution. Most SLAM approaches using the RBPF depend on a proposal that is derived from the motion model using the observations to offset the difference between the proposal and target distribution. The FastSLAM 2.0 algorithm [22] employs a proposal that also takes into account the most recent observation via the linearization of the motion and observation models allowing them to compute the proposal in closed form. In contrast, [8] uses a mixture proposal that combines hypotheses from the motion and observation models weighted according to either the motion or observation models (such a mixture proposal has been previously used for global localization and tracking [23], [24]). However, computing the particle weights according to two different models generates an inconsistent set of hypotheses that in [8] is not treated in a principled way.

We also use a mixture proposal generating hypotheses from both the motion model and a second distribution that depends on the latest observation and learned map thus far. We compute the weights correctly avoiding the problem described in [8].

Formally, our proposal distribution is given by:

$$q(s_t|s_{t-1}, z_t, u_t) = \\ \phi q_{global}(s_t|z_t, m_{t-1}) + (1-\phi) q_m(s_t|s_{t-1}, u_t) \tag{12}$$

where ϕ is known as the mixing ratio. Sampling from the motion model, $p(s_t|s_{t-1}, u_t)$, 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 $(J^T J)^{-1}$. In order to sample from q_{global}, we generate it by taking advantage of the $3D$ geometry of our map and camera configuration.

Specifically, let z_t be the current observation and $m_{t-1}^{(i)}$ be the map of the ith particle learned up to time $t-1$. Given the correspondences in z_t, we can compute the transformation that relates the two coordinate frames using weighted-least squares. The procedure is similar to that described in section III-C except that in this case instead of minimizing the re-projection error of the $3D$ points, we minimize their Euclidean distance in $3D$ [25]. The problem with this approach is that it is sensitive to outliers that are present in our data association. To overcome this problem, we employ a RANSAC type approach where we select subsets of the point correspondences and compute a candidate robot pose for each. We have found that generating as few as 200 candidate poses, $s_{t_{1\ldots200}}$, is sufficient for good localization given our noisy observations. For computational efficiency, we only compute the candidate poses with respect to the map of the most likely particle at time $t-1$.

In order to sample from q_{global}, we evaluate, using Equation 5, the probability of our latest observation given each candidate pose. We then fit a Gaussian distribution to these data points such that:

$$q_{global} = N_{global}(\mu, \Sigma)_{s_{t_{1\ldots200}}} \tag{13}$$

We compute the weight for the $i-th$ particle correctly by evaluating the ratio of the target and proposal distributions,

$$w_i = \frac{p(z_t|s_t'^{(i)}, m_{t-1}^{(i)}) p(s_t'^{(i)}|s_{t-1}^{(i)}, u_t)}{\phi q_{global}(s_t'^{(i)}|z_t, m_{t-1}^{(i)}) + (1-\phi) p(s_t'^{(i)}|s_{t-1}^{(i)}, u_t)} \tag{14}$$

Each of the distributions involved is a Gaussian that we have already described how to compute. One should notice that the weights for the particles are equal to the observation likelihood scaled by the ratio of the probability of the pose under the motion model and the weighted sum of the probability under the motion model and the *global* distribution. That is those particles that are supported by both models are given weights that are mostly proportional to the observation probability while those that disagree are penalized.

Finally, our empirical results show that using a constant mixing ratio tends to generate noisy robot trajectories. This is the result of the bad data associations in our observations. Also, the observations are often dominated by landmarks that were seen most recently, biasing the *global* distribution towards the most recently added landmarks. This potentially prohibits the closure of large loops. To correct this, we are varying the mixing ratio as a function of the ratio of old landmarks to total landmarks observed at time t. An old landmark in our case is defined as one that has not been observed for longer than 2 minutes. Using a variable mixing ratio, we rely on the standard proposal in the short term and the mixture proposal for closing loops. In the next section, we give the complete algorithm for our particle filter.

V. The $\sigma SLAM$ Algorithm

Algorithm 1 gives the filter update steps for $\sigma SLAM$. The input to the algorithm is the set of particles from the previous time step, the current observation and the visual odometry estimate. It starts by computing the ratio of old to total number of landmarks. If this number is larger than 30%, poses are sampled using the mixture proposal otherwise all poses are sampled from the motion model. In the latter case the particle weights are just proportional to the likelihood of the observation given by Equation 5. If the mixture proposal is used, then the weights are computed according to Equation 14. As we mentioned earlier, for efficiency, we only compute q_{global} with respect to the most likely particle from the previous time. If we do not wish to make this approximation then we must move the code that estimates q_{global} inside the last **for** loop, i.e., compute it for every particle.

Algorithm 1 $update(P_{t-1} = \{x^{(i)}, w^{(i)}\}_{(t-1)}^{i=1,\ldots,M}, z_t, u_t)$

if $\frac{old\ landmarks}{total\ landmarks} > 30\%$ **then**
  $use\_mixture = true$
  P_{t-1}^{ML} = most likely particle from P_{t-1}
  $\{h^{(j)}, h_w^{(j)}\}_{j=1,\ldots,N} = generatePoses(P_{t-1}^{ML}, z_t)$
  for $i = 1$ to $i = N$ **do**
    compute weight $h_w^{(i)} = p(h^{(i)}|z_t, m_{t-1}^{ML})$
  end for
  Estimate $N_{global}(\mu, \Sigma)$ from $\{h^{(j)}, h_w^{(j)}\}$
end if
{Compute the new set of particles}
if $use\_mixture == false$ **then**
  for $i = 1$ to $i = M$ **do**
    sample $s_t'^{(i)}$ from $p(s_t'^{(i)}|s_{t-1}^{(i)}, u_t)$
    compute $w_t^{(i)} = p(z_t|s_t'^{(i)}, m_{t-1}^{(i)})$
  end for
else
  for $i = 1$ to $i = M$ **do**
    Draw a random number, $rand$, from a uniform distribution
    if $rand < \phi$ **then**
      sample $s_t'^{(i)}$ from $p(s_t'^{(i)}|s_{t-1}^{(i)}, u_t)$
    else
      sample $s_t'^{(i)}$ from $N_{global}(\mu, \Sigma)$
    end if
    Compute $w_t^{(i)}$ using Equation 14
  end for
end if
Normalize particle weights
Resample
Update particle maps using z_t

VI. Experimental Results

We have implemented the $\sigma SLAM$ algorithm for our mobile robot seen in Figure 1. The robot is equipped with a PointGrey Research Bumblebee stereo vision camera mounted on a pan-tilt unit. The robot also has an on-board Pentium-based computer and no other functioning sensors.

For our experiments, we manually drove the robot inside two adjacent rooms, 10×16 meters, in our laboratory and around a large corridor. In both cases we collected the stereo images for offline processing. For the first data set, we followed a star-like exploration pattern starting from the middle of one room, exploring one corner at a time and returning back to the center to close the loop. We collected a total of 8500 frames and we processed all of them. The total length of the trajectory is 110 meters. For the corridor data set, we collected and processed 3400 frames for a trajectory of over 100 meters. The data we collected was challenging from a computer vision point of view because they include many frames with blurring caused by the uneven floor surface and people walking in front of the camera. Also the existence of large windows on the North, East and South walls of the rooms and at the four corners of the corridors cause large portions of some image frames to be saturated with bright light. Figure 2 shows a small sample of good and bad frames in our test sequences.

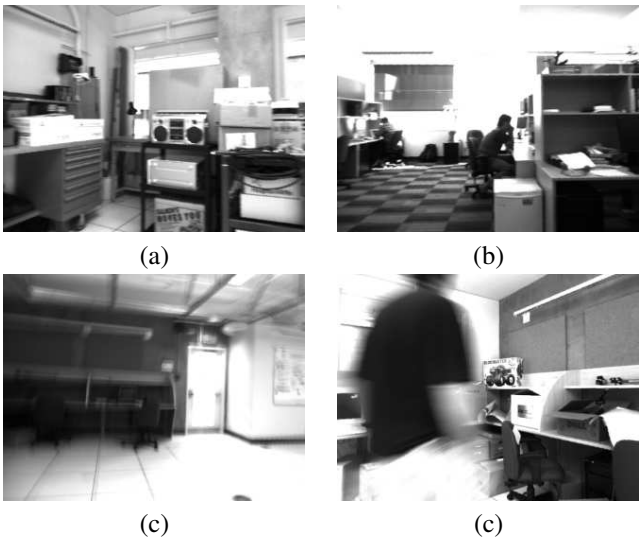


Fig. 2. Sample frames from our test sequence. A scene with (a) lots of texture that is good for localization; (b) excessive brightness and blurring caused by (c) uneven floor surface and a person walking in front of the camera

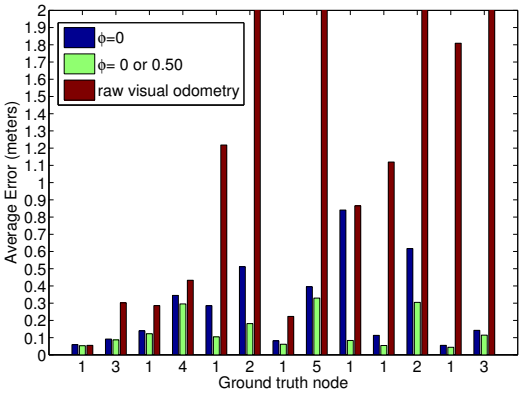


Fig. 3. The localization error for the filter using just the standard proposal, mixture proposal (with a variable mixing ratio) and the raw visual odometry estimates. Note that the robot starts at the left-most node in the plot and finishes at the right-most node.

Part (a) of Figure 4 shows a top-down view of the $3D$ landmark map, learned using $\sigma SLAM$ with 500 particles for our first data set. We used a mixing ratio that varied between 0 and 50% according to the age of the observed landmarks. The map shown is that of the most likely particle at the end of the trajectory. The map has a total of $80,000$ landmarks. Part (b) of Figure 4 also shows the occupancy grid associated with this particle, constructed using the method given in [5], and the dense stereo images returned by the stereo camera. The resolution of the grid cells is $15 \times 15cm$.

In order to measure the accuracy of $\sigma SLAM$, we have measured ground truth pose data for 5 different locations. The starting location is node 1 and each of the corners in the first

room are labeled nodes $2, 3, 4$ and 5. We have labeled these locations on the learned map shown in part (b) of Figure 4. The plot in Figure 3 shows the average position error at each node for the raw visual odometry, the particle filter using the standard proposal and $\sigma SLAM$ using the mixture proposal. One can see that the latter outperformed the other methods. In fact, the average error, over all nodes, using the mixture proposal is approximately $15cm$; using the standard proposal it is $25cm$ and using just the raw visual odometry measurements it is $159cm$. Also, using $\sigma SLAM$ the robot closed a large loop (25 meters long), going from node 1 to the room on the left and back to node 1 achieving an accuracy of approximately a factor of two better than using the standard proposal.

Part (c) of Figure 4 shows the 3D landmark map learned using the second data set and 500 particles. This map has approximately $22,000$ landmarks. For this experiment we also used a varying mixing ratio similarly to the first experiment but we only closed the loop once at the end of the trajectory. The error was less than $1m$ for an environment with very unreliable visual odometry due to the lack of texture and excessive light at the ends of the corridors. Part (d) of Figure 4 shows the occupancy grid corresponding to the most likely particle at the end of the trajectory.

Figure 5 shows the processing time for each frame of the two room image sequence running on a Pentium Xeon 3.2 GHz computer. The average time over all frames is 1.5 seconds. A large part of this, 0.35 seconds, is spend on SIFT extraction and matching. The time increases as the number of landmarks in the map increases because it takes longer to compute data associations, update the map for each particle and copy maps between particles. For efficiency, we are organizing all SIFT keys using a kd-tree data structure as originally suggested in [1]. In addition, in order to perform efficient copying of maps between landmarks, we have implemented the tree data structure introduced in [4]. We should emphasize that the latter data structure is only used to efficiently copy the $3D$ landmark maps and not the occupancy grids. We only construct the latter to demonstrate that we can. In fact, we could manage the grids efficiently by implementing the data structure introduced in [3] but we have not done this yet. There are frames when the processing time increases well above the average time; this is due to instances when a large number of particles and in effect their maps must be pruned during the resampling step because of their low likelihood.

VII. Conclusion

We have presented $\sigma SLAM$, an approach for vision-based SLAM using a stereo camera and the Rao-Blackwellised Particle Filter. We have proposed a specific model that does not use mechanical odometry information and operates using as the only input a video stream of stereo images. We derive our motion model from the sparse optical flow of image points identified using SIFT. We have proposed a novel mixture proposal distribution that allows us to robustly close large loops. We have shown that we can construct accurate metric

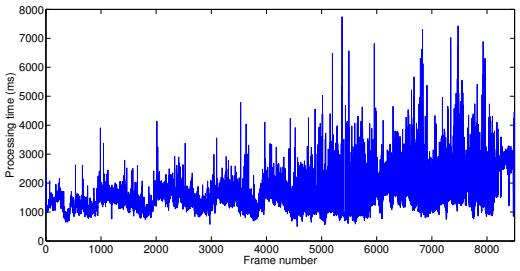


Fig. 5. Plot of the average processing time vs frame number for 500 particles using $\sigma SLAM$.

maps of $3D$ point landmarks and $2D$ occupancy grid maps from dense correlation-based stereo.

Finally, we should note, that our model is derived from the start for unconstrained $3D$ camera motion. In this paper, we have presented only results for planar robot motion for which we have ground truth data allowing us to evaluate its performance. However, we have evidence [23] that the system can solve for 6 degrees of freedom and we are currently exploring methods for evaluating it more precisely. Possible applications are a wearable stereo camera and the construction of volumetric representations of indoor spaces.

VIII. Acknowledgment

We would like to thank M. Griffin and A. Shyr for help with software development and N. de Freitas and K. Murphy for help with particle filters. This work was supported by NSERC, the IRIS Network of Centres of Excellence and the Canadian Space Agency.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Int. Conf. on Computer Vision*, Corfu, Greece, September 1999, pp. 1150–1157.

[2] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1985, pp. 116–121.

[3] A. I. Eliazar and R. Parr, "DP-slam 2.0," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*. New Orleans, LA: IEEE Press, 2004.

[4] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, 2004, to appear.

[5] V. Tucakov, M. Sahota, D. Murray, A. Mackworth, J. Little, S. Kingdon, C. Jennings, and R. Barman, "A stereoscopic visually guided mobile robot," in *Proc. of Hawaii International Conference on Systems Sciences*, 1997.

[6] S. Se, D. G. Lowe, and J. J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[7] T. D. Barfoot, "Online visual motion estimation using FastSLAM with SIFT features," Edmonton, Alberta, August 2005, pp. 3076–3082.

[8] N. Karlsson, E. D. Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Barcelona, Spain, April 2005, pp. 24–29.

[9] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. International Conference on Computer Vision, Nice*, Oct. 2003.

[10] S. Thrun, "Robot mapping: A survey," Carnegie Mellon university, Technical Report CMU-CS-02-11, February 2002.

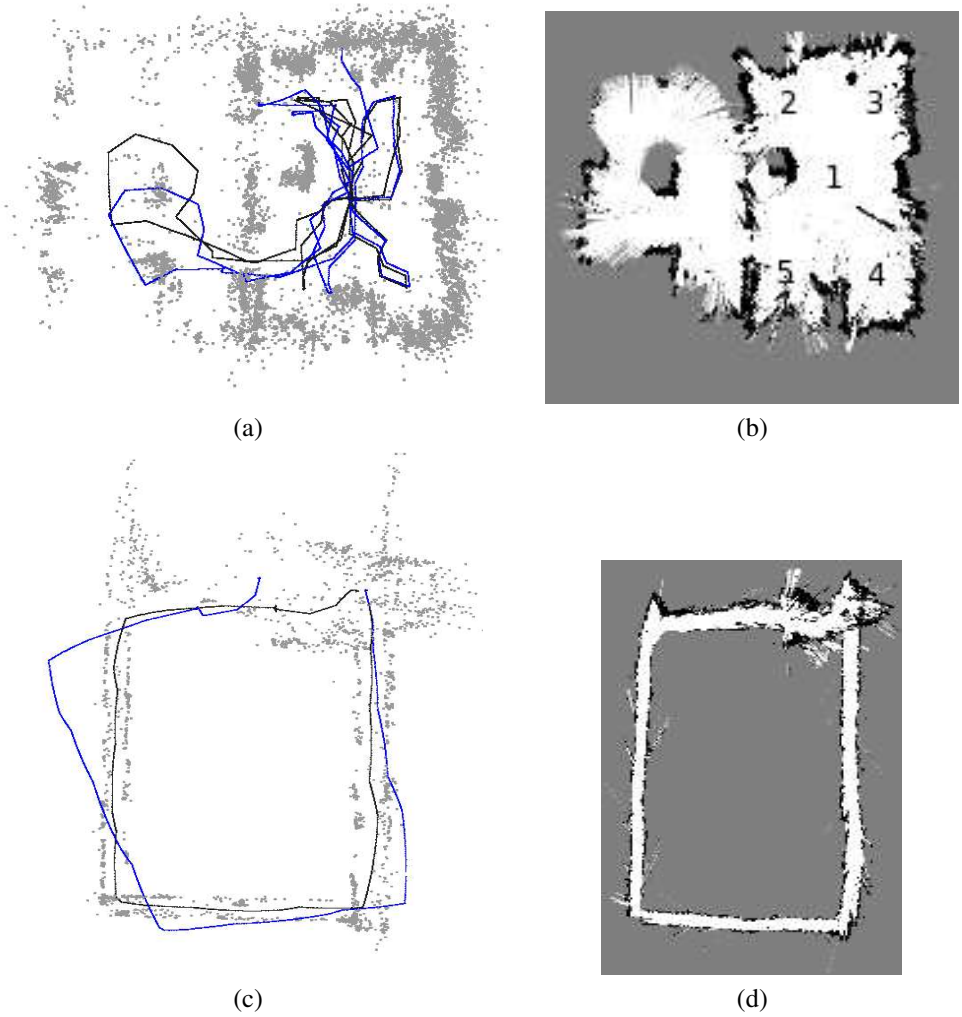


Fig. 4. Examples of the maps learned with $\sigma SLAM$ for (top row) two adjacent rooms size 16×10 meters and (bottom row) a corridor. Parts (a) and (c) show a top-down view of the maps of $3D$ landmarks used for localization. Shown in gray is the filter trajectory and in blue the raw visual odometry trajectory. Parts (b) and (d) show the occupancy grids constructed using the camera location estimated with $\sigma SLAM$. In the grid, white pixels represent empty space while black pixels represent occupied space. Gray pixels denote unobserved areas. The grid's resolution is $15 \times 15cm$.

[11] K. Murphy, "Bayesian map learning in dynamic environments," in *Neural Information Processing Systems (NIPS)*, 1999, pp. 1015–1021.

[12] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo in Practice*. Springer-Verlag, 2001.

[13] A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Uncertainty in Artificial Intelligence*, 2000.

[14] R. Sim, P. Elinas, M. Griffin, and J. Little, "Vision-based SLAM using the Rao-Blackwellised Particle Filter," in *IJCAI Workshop on Reasoning with Uncertainty in Robotics (RUR)*, Edinburgh, Scotland, 2005.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, UK: Cambridge Univ. Pr., 2000.

[16] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, p. 215, 2003.

[17] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, 2004, pp. 652–659.

[18] J. Campbell, R. Sukthankar, and I. Nourbakhsh, "Techniques for evaluating optical flow for visual odometry in extreme terrain," in *IEEE/RSJ Int. Workshop on Robots and Systems (IROS-04)*, Sendai, Japan, October 2004.

[19] A. Levin and R. Szeliski, "Visual odometry and map correlation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, Washington, DC, USA, 2004.

[20] A. Milella and R. Siegwart, "Stereo-based ego-motion estimation using pixel tracking and iterative closest point," in *In Proceedings of IEEE International Conference on Computer Vision Systems (ICVS 2006)*, Manhattan, New York, USA, 2006.

[21] 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.

[22] 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 *Proceedings of the Eighteenth Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*. San Francisco, CA: Morgan Kaufmann Publishers, 2003, pp. 1151–1156.

[23] P. Elinas and J. Little, "σMCL: Monte-Carlo localization for mobile robots with stereo vision," in *Proceedings of Robotics: Science and Systems*, Cambridge, MA, USA, 2005.

[24] S. Thrun and D. Fox, "Monte carlo localization with mixture proposal distribution," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2000.

[25] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 5, pp. 698–700, 1987.