Vision-based SLAM using the Rao-Blackwellised Particle Filter

Robert Sim, Pantelis Elinas, Matt Griffin, and James J. Little *Laboratory for Computational Intelligence*
University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
{*simra,elinas,mgriffin,little*}*@cs.ubc.ca*

ABSTRACT

We consider the problem of Simultaneous Localization and Mapping (SLAM) from a Bayesian point of view using the Rao-Blackwellised Particle Filter (RBPF). We focus on the class of indoor mobile robots equipped with only a stereo vision sensor. 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 reliable motion estimates. Landmark estimates are derived from stereo vision and motion estimates are based on visual odometry. We distinguish between landmarks using the Scale Invariant Feature Transform (SIFT). Our work defers from current popular approaches that rely on reliable motion models derived from odometric hardware and accurate landmark measurements obtained with laser sensors. We present results that show that our model is a successful approach for vision-based SLAM, even in large environments. We validate our approach experimentally, producing the largest and most accurate vision-based map to date, while we identify the areas where future research should focus in order to further increase its accuracy and scalability to significantly larger environments.

I. INTRODUCTION

Robot localization is a well studied problem as it is considered one of the most significant obstacles towards real robot autonomy. *Simultaneous localization and mapping* (SLAM) is the problem of estimating both the 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 the following parameters

- Sensors, i.e., sonar, laser or vision, wide or narrow field of view
- Map representation, i.e., occupancy grid, $2D$ or $3D$, natural or specialized landmarks
- Robot dynamics
- Environment dynamics, i.e., indoor or outdoor, static or dynamic
- Framework for combining over time the incoming sensor measurements and robot control signals

Each of these choices has advantages and disadvantages as well as direct implications on the applicability of the algorithms based on them. For example, sonar sensors are cheap and can provide information 360 degrees around the robot but have limited resolution and suffer from the perceptual aliasing problem more than any of the others. Laser sensors, and vision, on the other hand, most often have a limited field of view but laser sensors have high depth resolution while vision has high perceptual resolution. Omnidirectional cameras provide a 360 degree field of view but pay a high penalty on diminished image resolution. Sonar and laser sensors are better suited for planar robots that use $2D$ occupancy grids [1] to represent maps. Vision can be used to construct $2D$ [2] and $3D$ occupancy grids or maps of $3D$ natural landmarks [3] and it is not limited to planar robots. Environment dynamics are also important as most algorithms for indoor robots do not scale well for outdoor robots.

In terms of theoretical frameworks, the Extended Kalman Filter (EKF) has been the most common approach since its application by Smith, et al. [4]. By maintaining a covariance matrix which encompasses all landmarks this method allows the EKF to develop pose and landmark estimates incrementally. However as the number of landmarks grows this matrix quickly becomes difficult to expand and update efficiently. It is necessary to update all elements for any new observation and this leads to a complexity of $O(N^2)$, where N is the number of landmarks. The EKF is also very sensitive to outliers in landmark detection. A single outlier measurement once incorporated into the covariance matrix cannot be corrected at a later time if more information becomes available.

Another approach is the use of Particle Filters to approximate the posterior distribution over robot poses and maps. PFs can handle outliers better than the EKF but scale poorly with respect to the dimensionality of the state. The Rao-Blackwellised Particle Filter (RBPF) reduces this problem by factoring the state variables such that by sampling over a subset of them we can marginalize out the remaining ones [5]. Murphy [6] was the first to study the application of RBPFs to SLAM and others

followed [7]. Sampling over robot poses allowed him to independently estimate each landmark using an EKF. A naive implementation would yield a complexity of $O(MN)$, where M is the number of new particles at each step. Montemerlo et al. [8] present FastSLAM a variant of RBPF-based SLAM that introduces a tree-based structure which refines this complexity to $O(M \log N)$ by sharing landmarks between particles. They prove that FastSLAM converges to the correct map for special cases with a single particle. Similarly, Eliazar and Parr have constructed an efficient $2D$ occupancy grid representation for particle-based SLAM [9].

Central to both the EKF and PF approaches is a model of the evolution of the system is given. In SLAM this is traditionally a function of odometric hardware. They also require that a probabilistic model of observations is given. The parameters of this model are induced by our choice of sensor and map representation.

The approach we advocate here depends on a motion model based on visual odometry and an observation model based on $3D$ landmarks from stereo vision coupled with the Scale-Invariant Feature Transform (SIFT) detector [10]. SIFT is used for robust data association. These features are desirable as landmarks because they are invariant to image scale, rotation and translation as well as partially invariant to illumination changes and affine or 3D projection. This combination can result in many viable landmarks from an unaltered environment and in fact SIFT has been shown to outperform other leading edge image descriptors in matching accuracy [11].

The major contributions of this paper are two-fold. First, we present RBPF-based SLAM utilizing vision-based sensing, rather than traditional range sensing with a laser. Our motion model depends on visual odometry that generalizes to unconstrained $3D$ motion. That is, we assume no prior knowledge of the control actions that drive the camera through the world. Furthermore, where previous implementations of the SLAM algorithm have generally employed sensors with a wide field of view, our experimentation demonstrates the performance of the algorithm using sensors with a narrow field of view. We leverage the strengths of particle filter-based methods for uncertainty estimation (such as the possibility of multi-modal estimates), with data association techniques that were previously only applied to Kalman-filter based estimators [3, 12, 13].

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 provide experimental results to support the validity of our approach and conclude by discussing scalability issues and implementation pitfalls along with directions for future work.

II. SLAM USING THE BAYES FILTER

Formally, and in accordance with popular SLAM literature, let at time t, s_t denote the robot's pose, 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 as 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 1 we get [14]

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

where η is a normalizing constant.

Equation 2 allow 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 robot motion 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 we approximate it using a Particle Filter as described in the next section.

A. Rao-Blackwellised Particle Filters

In the previous section we showed how we can recursively estimate the posterior density $Bel(x_t)$ using the Bayes filter. As discussed we cannot directly implement this filter and so we employ a common approximation technique known as Particle Filtering. PF is a general method for approximating $Bel(x_t)$ using a set of m weighted particles, $Bel(x_t) = \{x^{(i)}, w^{(i)}\}_{i=1,\cdots,m}$. The system is initialized according to $p(x_0)$ and the recursive update of the Bayes filter proceeds in the following steps:

1) for each particle i
2) Sample from $Bel(x_{t-1})$ using the weighted samples, giving $\{x_{t-1}^{(i)}\}$
3) Sample from $q_t = p(x_t|x_{t-1}, u_t)$ (also known as the proposal distribution), giving $x_t^{(i)}$
4) Compute the importance weight, $w^{(i)}$ according to $p(z_t|x_t^{(i)})$, the observation density
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* [15]. 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 reduces the complexity of the PF by sampling over a subset of the *state* variables and marginalizing out the remain ones [16]. In the case of SLAM, we sample over possible robot poses, s_t, and then marginalize out the map [6]. The posterior then is factored as:

$$Bel(x_t) = Bel(s_t, m_t) = p(s_t, m_t|z^t, u^t) = p(s_t|z^t, u^t) \prod_k p(m(k)|s^t, z^t, u^t) \quad (3)$$

where $m(k)$ 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 the details of our approach to vision-based SLAM using the RBPF. 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.

A. Map Representation

We construct maps of naturally occurring $3D$ landmarks similar to those proposed in [3]. Each landmark is a vector $l = \{P, C^G, \alpha, s, f\}$ such that $P = \{X^G, Y^G, Z^G\}$ is a 3-dimensional position vector in the map's global coordinate frame, C^G is the 3×3 covariance matrix for P, and α, s, f describe an invariant feature based on the Scale Invariant Feature Transform [10]. Parameter α is the orientation of the feature, s is its scale and f is the 36-dimensional key vector.

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 (Figure 1). 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 [10]. 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 $o_j = \{P_{o_j} = \{X_{o_j}^L, Y_{o_j}^L, Z_{o_j}^L\}, p_{o_j} = \{r_{o_j}, c_{o_j}, 1\}, C^L, \alpha, s, f\}$ where $p_{o_j} = \{r_{o_j}, c_{o_j}, 1\}$ is the image coordinates of the point and $j \in [1 \cdots n]$.

Fig. 1. Sample left and right images from the stereo head.

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

$$P_{o_j}^G = R_{s_t} P_{o_j} + T_{s_t} \quad (4)$$

Using Equation 4 and the Mahalanobis distance metric we can define the observation log-likelihood, $\log p(z_t|m_t^i)$. 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:

$$\log p(z_t|m_t^i) = \sum_k \log p(o_k|l_k^i) \quad (5)$$

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

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

where the correspondence covariance S is given by the sum of the transformed observation covariance $C_{o_k}^L$ and

the landmark covariance C_k^G:

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

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 $\log p(z_t|m_t^i)$ can be such that raising it to the exponential to evaluate the i-th particle weight:

$$w_i = \frac{p(z_t|m_t^i)}{\sum_{j=1}^N p(z_t|m_t^j)} \quad (8)$$

results in zero weights. In order to preserve numerical accuracy, we note the following simplification. Let $H_i = -\log p(z_t|m_t^i)$. Without loss of generality, assume that m_t^0 is the particle that minimizes H_i. Then for all particles:

$$\log p(z_t|m_t^i) = -(H_0 + H_i'). \quad (9)$$

where $H_i' = H_i - H_0$.

Substituting into Equation 8:

$$w_i = \frac{\exp(-(H_0 + H_i'))}{\sum_{j=1}^N \exp(-(H_0 + H_i'))} \quad (10)$$

$$= \frac{\exp(-H_0)\exp(-H_i')}{\exp(-H_0)\sum_{j=1}^N \exp(-H_i')} \quad (11)$$

$$= \frac{\exp(-H_i')}{\sum_{j=1}^N \exp(-H_i')} \quad (12)$$

Note that for m_t^0, $H_i' = 0$, so we guarantee that at least one particle has a numerator of 1, above, and the denominator is at least 1.0. This approach effectively eliminates the probability mass associated with outliers that is common to all particles. It is also important to note that using this approach assures that all particles have comparable weights – every particle has the same number of input observations, and outliers are represented in the model on a per-particle basis. Hence, a particle with more outlier matches will have a lower weight than a particle with better data association.

C. Motion Model

An essential component to the implementation of RPBF 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. 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. 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 III-B, resulting in 2 sets of landmarks L_{t-1} and L_t. We compute the camera motion using a non-linear optimization algorithm minimizing the re-projection error of the $3D$ coordinates of the landmarks. We employ the Levenberg-Marquardt (LM) non-linear optimization algorithm [17]. We utilize the $3D$ coordinates of our landmarks and use the LM algorithm to minimize their re-projection error. 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

$$\tilde{s}_t^{i+1} = \tilde{s}_t^i - \chi \quad (13)$$

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

$$\epsilon = \begin{bmatrix} \epsilon_0^T \\ \epsilon_1^T \\ \vdots \\ \epsilon_k^T \end{bmatrix} = \begin{bmatrix} p_t^0 - K(RP_{t-1}^0 + T) \\ p_t^1 - K(RP_{t-1}^1 + T) \\ \vdots \\ p_t^k - K(RP_{t-1}^k + T) \end{bmatrix} \quad (14)$$

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

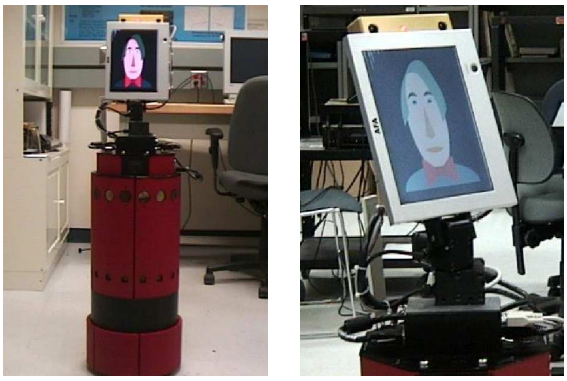
$$\begin{bmatrix} J \\ \lambda I \end{bmatrix} \chi = \begin{bmatrix} \epsilon \\ \lambda d \end{bmatrix} \Leftrightarrow (J^T J + \lambda I)\chi = J^T \epsilon + \lambda I d \quad (15)$$

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 [18] solving Equation 15, i.e., the normal equations, minimizes

$$||J\chi - \epsilon||^2 + \lambda^2 ||\chi - d||^2 \quad (16)$$

The normal equations can be solved efficiently using the SVD algorithm. A byproduct from solving Equation 16 is that we also get the covariance of the solution in the inverse of $J^T J$.

Fig. 3. Sample images from the 4000 frame sequence.



(a) (b)

Fig. 2. The RWI B14 robot we used for data collection as (a) seen from a distance and (b) closeup of its head.

IV. Experimental Results

For the purposes of our experiments, we used an RWI B14 robot with a BumbleBee stereo head from Point Grey Research. The robot is shown in Figure 2. We manually drove it through through two connecting rooms in a laboratory environment, and we collected 4000 images along a trajectory of approximately 67.5m. Figure IV displays a subset of the collected images (every 500th frame). While the visual odometry produces 6-DOF motion estimates, we chose to estimate only three parameters in constructing s_t from $\tilde{s}_t$. While for this particular experiment, this assumption was reasonable, we have preliminary results suggesting that relaxing the assumption altogether will be successful [19].

As a summary of the map construction process, Table I describes at 200 frame intervals the mean number of landmarks per particle (SIFT features observed more than three times), the total distance travelled according to the robot's odometer, and the total number of SIFT features (landmarks have been observed at least three times, whereas SIFT features have been observed at least once, and are removed if unobserved for a second time within three frames).

Figure 4 depicts the map constructed for the maximum-

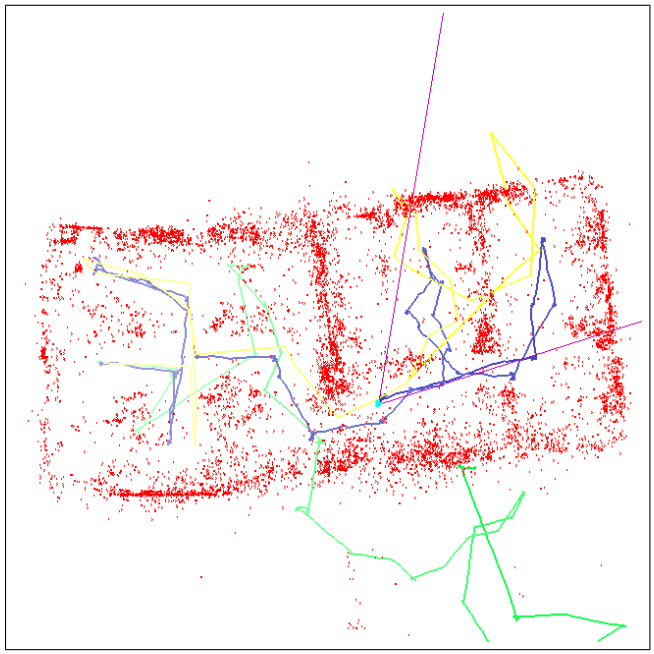


Fig. 4. The constructed map for the best sample at the end of exploration. The blue trajectory indicates the trajectory of the best sample and the green trajectory indicates the visual odometry measurements. The robot odometer (not used for map estimation) is plotted as a yellow trajectory. Landmark positions are marked with red 'X's. The set of particles is shown by the cyan blob near the center. The width of the map is approximately 18m.

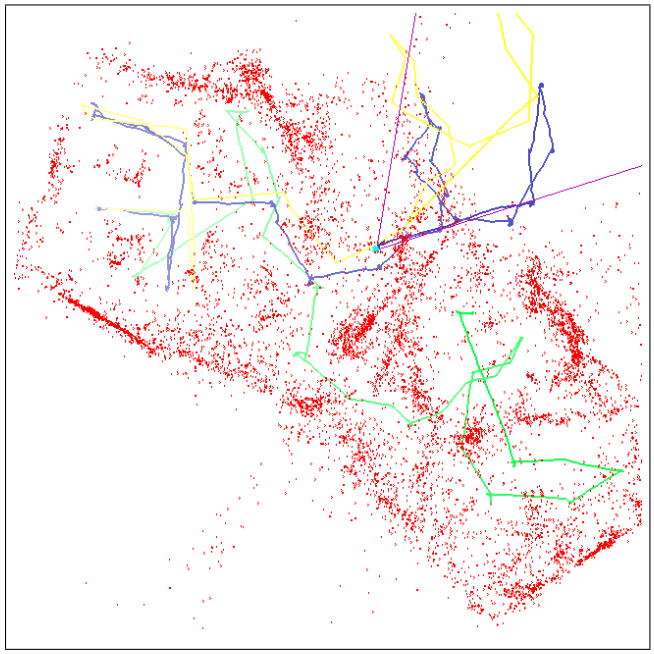


Fig. 5. The constructed map based solely on the visual odometry.

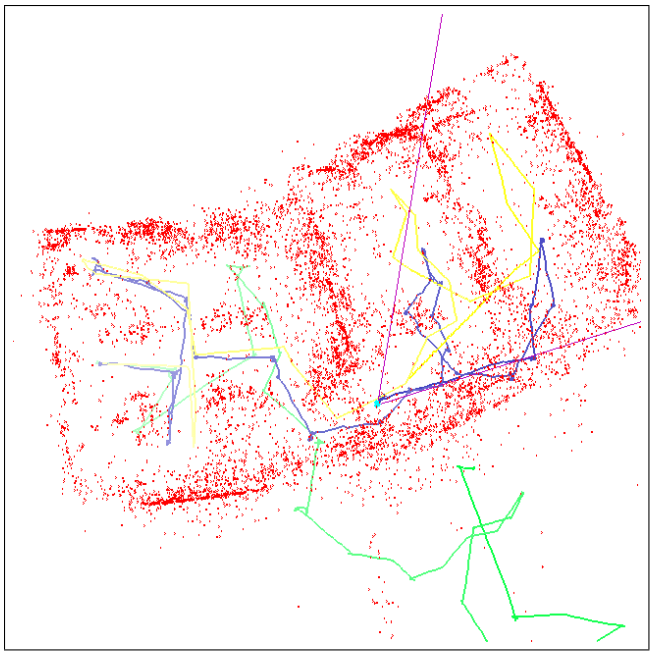


Fig. 6. The constructed map based solely on the robot's odometer (which was not used for constructing the map in Figure 4).

likelihood particle at the end of exploration. This map is not post-processed to remove noise or perform any global optimization. The blue trajectory indicates the trajectory of the best sample and the green trail indicates the visual odometry measurements. The robot odometer (not used for map estimation) is plotted as a yellow trajectory. All three trajectories begin from the origin, on the left side of the image. Landmark positions are marked with red 'X's. The set of particles is shown by the cyan blob near the center. To the best of our knowledge, this is the largest and most accurate SLAM-based visual map construction in existence. Similarly, Figures 5 and 6 depict the map as constructed using only visual odometry and the robot's odometry, respectively. Clearly, the filter out-performs both kinds of odometry. Further results and discussion can be found on the web at http:\\www.cs.ubc.ca/~simra/lci/fastslam/.

Figure 7 depicts the computation time for each frame of the sequence on a 2.6GHz Pentium 4 CPU. The mean compute time per frame is 11.9s. The base-line cost (horizontal line near about 2s) corresponds to the motion estimation, whereas the larger costs correspond to RBPF updates (which are triggered only when sufficient motion is detected). A major contributor to the increased cost over time is the cost of matching SIFT features. For this experiment, to ensure robustness in data association, we employed a linear-time comparison of image features with

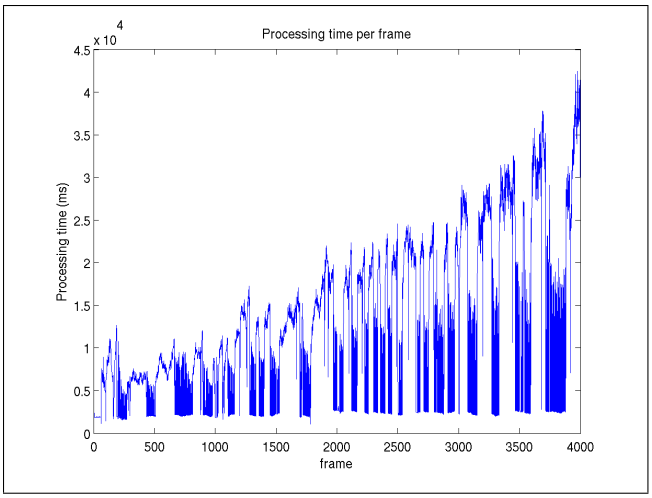


Fig. 7. Processing time per frame. The mean is 11.9s.

SIFT features in the map ($O(MN)$ where M is the number of observed features and N is the number in the map). There are a variety of fast methods for improving this result, particularly kd-trees. We have found that there is some degradation in data association quality using kd-trees, and that the kd-tree can become overcrowded over time as a result. Future work will address these issues.

V. Scalability Issues and Implementation Pitfalls

In this paper, we have presented experimental results which push the envelope for what can be accomplished using vision and no prior knowledge of the camera's motion. In particular, we are successfully building accurate maps over long-range motion. However, there are several considerations that were taken into account in order to compute an accurate result in a reasonable amount of time.

There are two barriers to full frame-rate operation. First, the number of particles must be small in order to update the maps in a reasonable amount of time. While some papers have argued that a proposal distribution conditioned on the observation can lead to a filter that converges with only one particle, we would argue that this distribution is highly over-confident and somewhat biased, necessitating the injection of noise into the distribution, and also necessitating a reasonably large number of particles to ensure diversity in the filter (particularly important for loop closing). For these experiments, we used 400 particles, and we believe the loop can be reliably closed over reasonable distances[1]. We have not experimented significantly with fewer particles, or the level of noise that must be injected.

[1]Note, that with a limited field of view, the filter must also close loops when the camera rotates away from and back to previous viewing directions.

TABLE I

Map construction summary

Frame Number	Mean Landmarks per particle	Distance traveled (m)	SIFT Features
4000	11085	65.71	38394
3796	10595	61.94	36384
3600	10056	58.82	34404
3400	9694	54.83	32874
3199	9065	52.05	30639
3000	8608	48.64	28719
2797	8355	43.58	27369
2600	7769	41.02	25359
2400	7155	38.07	23259
2200	6689	34.37	21594
1998	6213	31.11	19869
1800	5379	29.25	17259
1600	4857	26.01	15264
1396	4353	22.22	13254
1200	3773	18.69	11424
997	3103	14.98	9399
800	2625	11.48	7674
600	2206	6.87	6264
400	1542	4.02	4194
191	812	0.62	2109

The second barrier is the management and correspondence of SIFT features. We use 36-element SIFT feature vectors, but perform a list traversal to match each feature. As mentioned above, matching can be improved by using a kd-tree, but this can present additional complications for key maintenance (for example, deleting unmatched keys from the tree after a few frames). Without sophisticated key maintenance, the tree can become over-populated, making it very difficult to verify good matches. The rate at which SIFT keys are added is another consideration, and we insert a limited number of keys into the database at each frame (10-15). These insertions are predicated on the new keys being sufficiently distinct from the keys already in the database. Without these limits, the number of SIFT keys can grow by up to 500 keys per frame.

Currently, we compute 6-DOF motion estimates, but represent only a 3-DOF robot position. In related work, full 6-DOF SLAM has been successful in only very small environments [12, 20], and our future work will address this challenging problem. The holy grail of the SLAM community would be to accomplish large-scale 6-DOF SLAM with only a monocular camera. We are also considering how our work could be extended to that problem. We believe that the filter would require a larger number of particles but that in principle only minor changes would be required to our system.

VI. Conclusions

In this paper we have presented our model for vision-based SLAM from a Bayesian point of view using the RBPF. We show that we can successfully construct dense

metric maps of $3D$ point landmarks for long camera trajectories in the order of 68 meters and 4000 image frames. We have utilized SIFT for identifying landmarks and defining the observation function of our model. We diverged from popular SLAM literature by not relying on motion estimates based on odometric hardware but only on visual odometry. We have identified a number of areas that need further work to increase the computational efficiency, and representational power of our method, in order to build accurate maps of even larger environments.

REFERENCES

[1] 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.

[2] 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.

[3] S. Se, D. G. Lowe, and 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.

[4] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Workshop on Spatial Reasoning and Multisensor Fusion*, 1987.

[5] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[6] K. Murphy, "Bayesian map learning in dynamic environments," in *1999 Neural Information Processing Systems (NIPS)*, 1999. [Online]. Available: citeseer.ist.psu.edu/murphy00bayesian.html

[7] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the CONDENSATION algorithm for robust, vision-based mobile robot localization," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*. Ft. Collins, CO: IEEE Press, June 1999, pp. 2588–2593.

[8] 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 Conf. on Artificial Intelligence*. Edmonton, Canada: AAAI, 2002.

[9] A. I. Eliazar and R. Parr, "DP-slam 2.0," in *Proc. ICRA 2004*. New Orleans, LA: IEEE Press, 2004.

[10] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Int. Conf. on Computer Vision*. Corfu, Greece: IEEE Press, September 1999, pp. 1150–1157.

[11] 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

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

[13] M.-H. Y. Jason Meltzer, Rakesh Gupta and S. Soatto, "Simultaneous localization and mapping using multiple view feature descriptors," Sendai, Japan, September 2004.

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

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

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

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

[18] 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.

[19] P. Elinas and J. Little, "σmcl: Monte-carlo localization for mobile robots with stereo vision," in *(in submission) Proceedings of Robotics: Science and Systems*, 2005.

[20] A. J. Davison and N. Kita, "3D simultaneous localisation and map-building using active vision for a robot moving on undulating terrain," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, Lihue, HI, December 2001, pp. 384–391.