Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Autonomous vision-based exploration and mapping using hybrid maps and Rao-Blackwellised particle filters

Robert Sim and James J. Little
Department of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, BC, V6T 1Z4
{simra,little}@cs.ubc.ca

Abstract— **This paper addresses the problem of exploring and mapping an unknown environment using a robot equipped with a stereo vision sensor. The main contribution of our work is a fully automatic mapping system that operates without the use of active ranger sensors (such as laser or sonic transducers), can operate in real-time and can consistently produce accurate maps of large-scale environments. Our approach implements a Rao-Blackwellised particle filter (RBPF) to solve the simultaneous localization and mapping problem and uses efficient data structures for real-time data association, mapping, and spatial reasoning. We employ a hybrid map representation that infers 3D point landmarks from image features to achieve precise localization, coupled with occupancy grids for safe navigation. This paper describes our framework and implementation, and presents our exploration method, and experimental results illustrating the functionality of the system.**

I. Introduction

For a robot to operate autonomously in its environment, it first requires an accurate representation that facilitates localization and navigation. While the problem of automatically constructing such a representation is largely solved for robots equipped with active range-finding devices (and generally operating in planar worlds, e.g. [1]), for a variety of reasons the task remains challenging for robots equipped only with vision sensors. This paper presents a solution to the autonomous exploration and mapping problem using a robot equipped only with a stereo camera and an odometry sensor. In particular, we demonstrate a consistent, convergent simultaneous localization and mapping (SLAM) solution and the generation of a map representation that facilitates both accurate localization and collision-free navigation. Furthermore, the mapping is accomplished in real-time under fully autonomous planning and control. In this light our work is unique among the extant vision-based mapping frameworks.

The last decade of robotics research has generated a multitude of approaches to the SLAM problem. Central to this problem is the probabilistic estimation of a map conditioned on a robot's noisy actions and observations. In general, it has been demonstrated that with a highly accurate sensor and some straightforward assumptions about the world (e.g. a planar pose space), a robot can successfully map a large indoor environment in real-time. The main difficulty with these approaches is that they rely on active (energy-emitting) laser range-finding sensors, and they assume that all of the important obstacles in the world lie in the plane of the sensor. One can easily demonstrate that many, perhaps most, environments contain obstacles violating this assumption. Furthermore, the data returned by a laser tends to be impoverished in that substantial travel may be required (integrating measurements along the way) in order to infer a robot's position. Finally, there are many potential scenaria where an active sensor is undesirable.

The primary alternative to active range sensing is the passive approach afforded by stereo imagery. These approaches benefit in that the information contained in a single image can often provide substantial discriminative power for localization, and that a stereo sensor can provide a 3-dimensional (or 2.5-D) representation of potential obstacles. The main drawback of stereo sensing and image-based sensing in general is that noise plays a substantial role in diluting a robot's inferential power, particularly as it applies to geometric reasoning. As a result, the application of successful range-based techniques has proven to be a challenge, particularly as it applies to occupancy-grid-based SLAM. This fact is evident in that of all of the vision-based SLAM solutions to date, the vast majority have computed only landmark (feature)-based representations and depended on human control or active range sensing for planning and obstacle detection and avoidance.

The goal of our work, therefore, is to use vision to autonomously explore an unknown environment and build a consistent map with an accuracy that is competitive with active range-sensing solutions. We aim to overcome the limitations of previous vision-based approaches by providing a measure of occupancy of the world to facilitate obstacle avoidance and navigation, and similarly we demonstrate that our approach can map obstacles that a traditional range finder would omit.

Our approach to the autonomous mapping problem is based on the successful combination of techniques in SLAM, efficient data association, data management and planning. In

1-4244-0259-X/06/$20.00 ©2006 IEEE

particular, our SLAM approach is based on mapping 3D point landmarks using the Rao-Blackwellised particle filter (RBPF) [2]. The landmarks are detected in images using the scale-invariant feature transform (SIFT) [3], matched using efficient best-bin first KD-tree search [4], and efficiently stored using the FastSLAM framework [5]. While in previous work we have demonstrated consistent, accurate SLAM results using this framework [6], [7], a landmark-based map is insufficient for path planning and obstacle avoidance. As such, we use a unique just-in-time occupancy representation, computed as a by-product of the SLAM filter in order to provide a reliable 2.5D spatial representation of the world. Using this representation, we perform real-time path planning and we employ a technique similar to frontier-based exploration [8] to ensure coverage of the environment.

The remainder of this paper presents related work, our approaches to SLAM, representation, and exploration, and finally experimental results demonstrating the success of our approach.

II. Related Work

This paper is an extension of our previous work addressing vision-based SLAM [6], [7], in which an RBPF framework was employed to perform SLAM on data sets collected while navigating a robot using human teleoperation. This paper extends our prior work by considering the problems of autonomous navigation and exploration, and presenting an efficient hybrid mapping approach for facilitating real-time occupancy updating. Our work is also distantly related to that of [9], in which a robot navigates and maps a small environment.

While the SLAM literature is vast, only a small number of solutions facilitate real-time control *and* consistent mapping. Furthermore, most of the SLAM literature focuses on active range sensing, such as the SICK laser range-finder. For brevity, we will consider here only those solutions that use vision or that present results directly relevant to our work.

Our SLAM solution is based on the Rao-Blackwellised particle filter approach presented by Murphy [2], and later popularized by a series of papers by Montemerlo *et al.* [5], [10]. In this approach, a set of samples are maintained representing the probability distribution over the robot's trajectory, and the map features (landmark or grid cell) probability estimates become independent when conditioned on the sampled trajectory. The RBPF approach has been successfully applied using vision-based landmarks, as in [7], and also using occupancy grid representations, as in [11], [12] and [13]. Other vision-based SLAM approaches include the view-based information filter by Eustice *et al* [14], and the landmark-based approaches by Davison and Kita [15], [16] and Se *et al* [17]. In the latter approaches, the environments considered were reasonably small, and landmark estimates were considered to be independent under a single trajectory hypothesis.

This paper also considers the problem of exploration for map construction. This problem has received considerable attention in the context of both landmark-based mapping [18], [19] and occupancy-grids [8], [20], [21]. In general, all of these approaches have in common the concept of information gain- which destinations in the world are most informative for mapping? While Stachniss *et al* demonstrate a full Bayesian computation of informative actions for exploration using the RBPF [21], we use the decidedly simpler approach suggested by Yamauchi, *et al* [8].

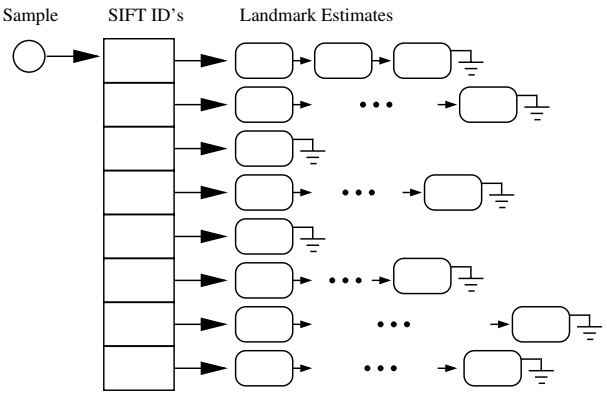

Fig. 1. Conceptually, each sample has an associated map, organized by SIFT descriptor. Each SIFT descriptor might have multiple landmark estimates, each spatially distinct.

III. Simultaneous Localization and Mapping

We represent map estimation as the evolution of a Rao-Blackwellised particle filter [2]. In this context, the trajectory and landmark distribution is modeled as a dynamic Bayes network, where trajectories are instantiated as samples, and the landmark distribution can be expressed analytically for each trajectory. At time t, let $s^t = \{s_1, \ldots, s_t\}$ denote the vehicle trajectory, 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 vehicle'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

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

It has been demonstrated elsewhere that $p(s^t, m_t | z^t, u^t)$ can be approximated by factoring the distribution in terms of sampled trajectories s^t, and independent landmark distributions conditioned on the sampled trajectories [2]:

$$p(s^t, m_t | z^t, u^t) \approx p(s^t | z^t, u^t) \prod_k p(m_t(k) | s^t, z^t, u^t) \tag{2}$$

where $m_t(k)$ denotes the $k-th$ landmark in the map. That is, we instantiate a set of samples s^t, propagate them according to u^t, and construct maps for each according to z^t.

A simplistic approach to running an RBPF for SLAM yields a storage complexity of $O(NK)$, where N is the number of samples at each step and K is the number of landmarks. However, Montemerlo et al. introduced a reference-counted binary search tree (FastSLAM) data structure which reduces this complexity to amortized $O(N \log K)$ by sharing landmark

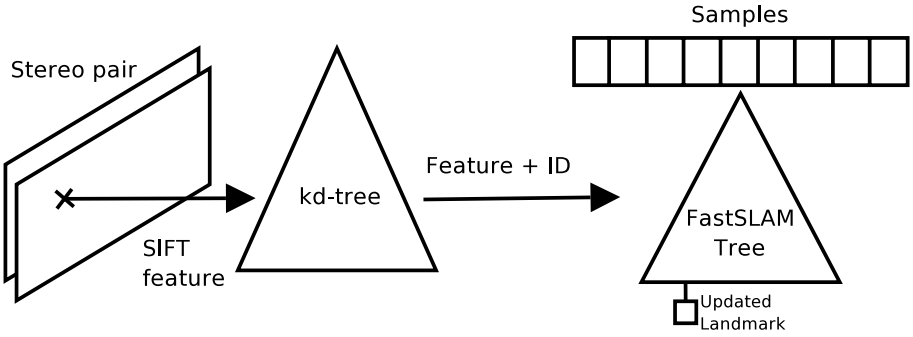

Fig. 2. Observation update (refer to text for details)

estimates between samples [5]. Each sample in the filter will share unaltered landmark estimates with other samples (those landmarks that have not been observed since the time the samples became siblings). Each landmark observation results in a landmark being copied and updated but the rest of the map remains unaltered.

Our landmark map employs the FastSLAM data structure with modifications to support vision-based data association. Conceptually, each sample has an associated set of landmark estimates, described by 3D Gaussian distributions, comprising its map. We take advantage of the descriptive power of the SIFT transform to improve the quality of data association. Each SIFT ID is mapped to a unique integer which is used to index into each sample's map. The node returned by querying a sample for a particular SIFT ID contains a linked list of landmark estimates – landmarks that are visually similar (sharing the same or similar SIFT descriptors), but which are spatially distinct, as shown in Figure 1. Individual landmark estimates are updated using the Extended Kalman Filter. The observation model is described further below.

IV. Observation model

Algorithm 1 Observation update procedure

F:= Extract SIFT keys and positions $f = \{k, p\}$ from image.
for all features f in F **do**
  id:= kd_tree_lookup($f.k$) {Index into kd-tree.}
  for all Samples s **do**
    List L:= s.map_lookup(id)
    Find most likely landmark estimate l in L, given $f.p$ {Maximizing observation likelihood.}
    Copy l if necessary {If shared with other samples.}
    Update l with $f.p$ using Kalman Filter update.
    Update w_t for s according to observation likelihood.
  end for
end for

Figure 2 and Algorithm 1 summarize the observation update process. We extract points of interest using the difference of Gaussian detector described in [3], and construct a 128-dimensional SIFT feature descriptor for each point. We subsequently perform a linear search of the keys in the left stereo image for the best match to each key in the right, subject to epipolar constraints, and determine its 3D position and covariance according to the well-known stereo equations:

$$Z = fB/d, \quad X = uZ/f, \quad Y = vZ/f \tag{3}$$

where f is the focal length of the camera, B is the base-line of the stereo head, d is the disparity between SIFT keys in the left and right images and $[u\ v]$ is the pixel position of the key in the right camera.

In addition to obtaining a 3D position estimate for each key, we compute the associated covariance matrix C, first by assuming fixed noise parameters σ_u, σ_v and σ_d for u, v and d respectively, and transforming the diagonal measurement covariance $S = diag(\sigma_u^2, \sigma_v^2, \sigma_d^2)$ according to the Jacobian ∇h of Equation 3:

$$C = \nabla h S \nabla h^T \tag{4}$$

In our experiments we typically use $\sigma_u = \sigma_v = 10.0$ pixels and $\sigma_d = 0.5$ pixels.

Once landmark observations are extracted from the stereo pair the landmark estimates must be updated for the individual samples. To efficiently store and access what can quickly become a large number of SIFT keys we use a best-bin-first KD search tree. The KD-tree facilitates nearest-neighbor matching in time logarithmic in the size of the tree, and has been demonstrated to be reliable for object recognition tasks [4]. The disadvantage of using a KD-tree is that it can sometimes produce not the nearest match but a close match. We maintain a single tree for the sensor and associate an arbitrary integer ID with each SIFT identifier we add. New keys are considered to be *candidate keys* and are not passed as an observation to the particle filter until they have been observed for a sufficient number of frames.

As described above, each particle's map is indexed by a set of IDs associated with SIFT descriptors and each indexed node contains a linked list of 3D landmarks sharing that descriptor. Multiple data associations can be entertained by the filter because each particle determines the specific landmark to which an observation corresponds.

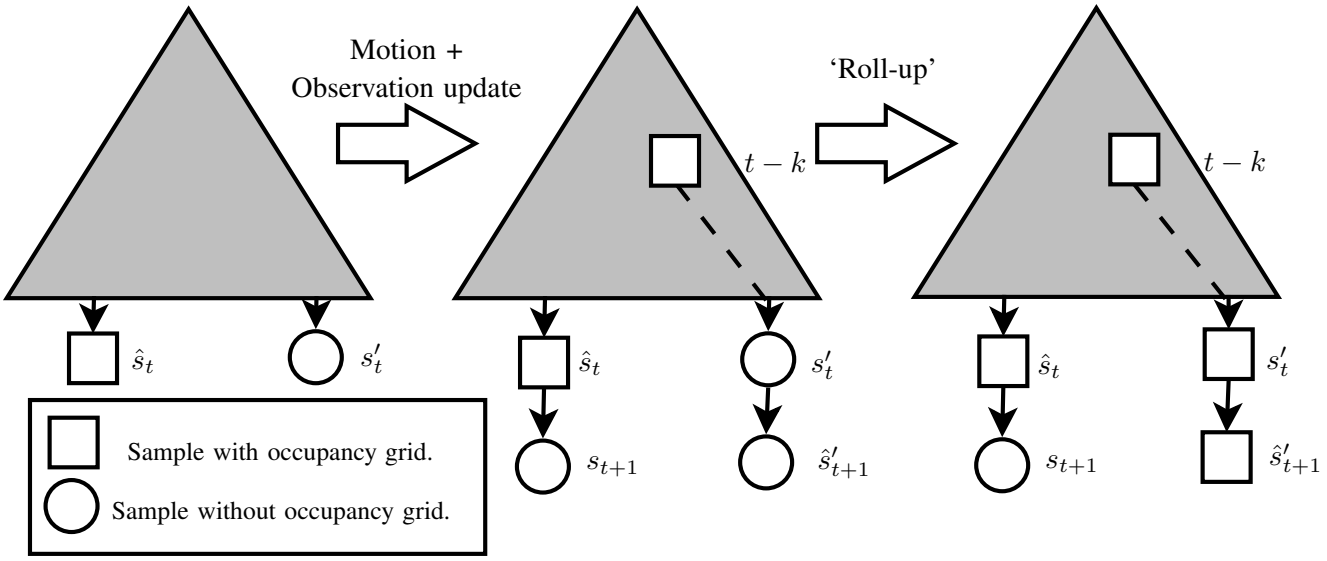

Fig. 3. Roll-up procedure for occupancy updating. The triangle represents the ancestry tree for samples in the RBPF. Squares represent samples with current occupancy grids, whereas circles represent samples without. When the ML sample switches from s to s' (indicated by the hat symbol), the occupancy grid is updated by 'rolling up' observations from time $t-k$ (the most recent ancestor with an up-to-date grid) to the present. Note that if s'_{t+1} is the only descendent of s'_t, reference counting will delete the grid associated with s'_t, and a similar check is performed on the ancestor at time $t-k$.

After determining the data association of an observation z_t to a landmark instance, the sample weight for that landmark is updated according to the observation likelihood:

$$w_{i,t} = \frac{p(z_t|s_i^t, m_{i,t-1})p(s_i^t|s_i^{t-1}, u_t)}{q(s_i^t|s_i^{t-1}, u_t, z_t, m_{t-1})}\ w_{i,t-1} \quad (5)$$

$$= p(z_t|s_{i,t}, m_{i,t-1})\ w_{i,t-1} \quad (6)$$

$$= k\exp(-0.5\Delta z^T\Sigma^{-1}\Delta z)\ w_{i,t-1} \quad (7)$$

where we are assuming a standard proposal distribution $q(s_i^t|s_i^{t-1}, u_t, z_t, m_{t-1}) = p(s_i^t|s_i^{t-1}, u_t)$, based on a model of robot odometry, and $\Delta z = h(s_{i,t}) - z_t$, $h(\cdot)$ is a generative model of the observation as a function of pose, Σ is the sum of the measurement covariance and prediction covariance. In short, the particle is weighted according to how well the current observation is consistent with the map constructed from that particle's trajectory. For a deeper discussion of the weighting and resampling approach, and a presentation of other proposal distributions of interest, we refer the reader to [7].

V. Just-in-time occupancy updating

Up to this point we have described the inferential system we use for solving the SLAM problem. Our SLAM solution is based entirely on inference from 3D landmark observations, and the map maintained for SLAM is likewise a landmark representation. However, in order to navigate, the robot also requires an occupancy representation. Our approach to this problem is to compute the maximum-likelihood occupancy grid after each observation update in the filter, conditioned on the current maximum-likelihood trajectory in the RBPF.

Consider for the moment an RBPF with only a single sample. After each observation update, the maximum-likelihood occupancy map can be computed by merging the occupancy map from the previous time-step with depth information from the current stereo image [9]. Specifically, assuming that the columns of the stereo image have been rectified to be perpendicular to the ground plane, for each column in the image we determine the closest obstacle by scanning from the bottom row to the top, computing depth from the stereo disparity and omitting obstacles which are situated above the robot's height. Given this radial representation of the robot's view, the maximum-likelihood occupancy grid is updated, conditioned on the sample's pose.

In the case of an RBPF with multiple samples, a difficulty arises if the maximum-likelihood state switches from one sample to another. A naive solution would be to maintain an occupancy grid for every sample, resulting in very high memory consumption and computationally expensive updating. Alternatively, one could use the DP-SLAM framework to efficiently maintain the distribution over all grids [13]. However, we are interested only in computing the most likely grid for the purposes of path planning, and so such a representation is computationally unnecessary. Instead, we propose a 'just-in-time' approach to grid updating.

When the maximum-likelihood (ML) sample switches from one sample $\hat{s}_t$ to another $\hat{s}'_t$, due to an observation update, the previous ML grid is stored with the old sample. The new ML sample checks its grandparent ($parent(parent(s'_t))$) to see if it was the ML sample at time $t-2$ (and hence had a grid), and if so, it copies that grid and updates it with the most recent depth image[1]. If the grandparent didn't have an updated grid, the search recurses backwards in time to $t-3$, *et cetera* to find

[1]By definition, if the current sample's parent has a grid then it was the ML sample at $t-1$.

the most recent ancestor in time with a grid, and then that grid is copied and updated by 'rolling up' the depth observations from that point in time to the present.

This system computes the ML grid, but a side-effect will be a large number of grids in memory. This problem is solved using a reference counting scheme. When a child sample 'rolls up' its parent's grid, the child decrements a reference count in its parent indicating that it no longer needs the parent's grid. Likewise, if a child sample is deleted due to resampling (or pruned because it no longer has descendents), it decrements the reference count stored by the parent. When the reference count stored by the parent reaches zero, it knows that none of its descendents depend on its grid and it can be deleted. Figure 3 summarizes this approach.

In a test experiment running an RBPF with 1000 samples for 8500 odometric measurements, 1880 observation updates, and 464 filter resampling operations, we observed that on average, 10 occupancy grids were stored in memory, and the maximum number at any point in the filter evolution was 23. Furthermore, we have observed that the cost of rolling up a grid to the current time step is usually very inexpensive (typically faster than other costs, such as SIFT extraction), to the extent that we can easily run the RBPF with 1000 samples and grids of size 700 by 700 pixels at 10cm resolution without significant concerns about computational cost or memory consumption.

It should be emphasized that we are not duplicating the DP-SLAM approach of Eliazar and Parr [13]. In that work the occupancy grid is used for localization and inference in the SLAM filter, whereas we do not use the grid for localization, nor do we maintain a full distribution over occupancy grids but rather compute only the maximum likelihood grid solely for the purposes of path-planning. In this sense, the grid is a by-product of the SLAM filter which computes pose and landmark estimates.

VI. Autonomous Exploration

The purpose of exploration is to construct a map such that a robot can navigate reliably throughout the environment. We use a frontier-based approach similar to that in [8] to compute optimal destinations in the world to expand the coverage of the map. Once full coverage is achieved, the robot may continue to explore regions that appear to be poorly mapped. We compute destinations using the ML occupancy grid $\hat{m}_t$. Specifically, the robot selects its next goal pose by evaluating safely traversable grid cells according to the value function

$$goal = \arg\max_s H(s|\hat{m}_t)$$

where

$$H(s|\hat{m}_t) = \sum_\theta V(s, \theta, \hat{m}_t)$$

and the value function V is computed by casting a ray from pose s in the direction θ in map $\hat{m}$, and returning a value dependent on the first non-empty grid cell s' that the ray intersects:

$$V_{s' \ occupied} = 0$$

and

$$V_{s' \ unknown} = exp(-0.5(||s' - s|| - r^*)^2/\sigma^2).$$

Here, V is maximized when the robot is a distance r^* from an unknown cell. In our current implementation, H samples angles at five degree intervals, r^* is 2.0m and σ is 0.5m. The purpose of r^* is to ensure that goal poses are not prematurely discovered to be occupied by obstacles, and to maintain a reasonable viewing distance for the landmark-based RBPF.

Once a goal pose is selected, the robot plans a path to it using A^* search in the occupancy grid and begins executing the trajectory, updating the RBPF along the way. After each observation update, the robot checks that its current plan is still safely traversable- if not, a new plan is computed, and if no such safe plan exists the robot selects a new goal.

Upon reaching a goal pose the robot pans its camera through 360 degrees to maximize the coverage of the map at that location, and upon successful completion of this manoeuvre it computes a new goal. Figure 4 depicts a small sequence of goal planning and navigation operations. A full animation of the exploration and map construction sequence is available on-line at `http://www.cs.ubc.ca/~simra/iros06/gridmap.avi`.

It is important to note that we have made several strong assumptions using this approach. First, path planning using only the ML occupancy grid can result in failed plans when the ML grid switches between samples. Our empirical experience indicates that while new paths must often be computed (and can be recomputed quickly and inexpensively), it is relatively rare that a goal is unreachable. Second, we are assuming that coverage in the occupancy grid is equivalent to good coverage (and good convergence) in the landmark-based map. Our only evidence to support this assumption is the fact that total localization failures (filter divergence) were extremely rare throughout our testing and experimentation. In short, while the approach we are using is sub-optimal from an information-theoretic standpoint, it is very easy to implement, fast to compute, produces consistent maps and achieves coverage.

VII. Experimental Results

For the purposes of our experiments, we used an Activmedia Powerbot robot with a Digiclops (trinocular) stereo head from Point Grey Research. The robot explored a laboratory environment consisting of two rooms of total size approximately 19m by 16.5m. Some example images of this environment are shown in Figure 5. The environment is particularly challenging due to the prevalence of natural light (causing image saturation), untextured specular regions (such as book cases and even the floor), variations in motion noise due to transitions between carpet and tile, and finally the dynamics of human occupants in the lab.

In this experiment, the robot collected 11,315 odometry measurements at a rate of about 5.4Hz. The maximum translation velocity of the robot was 0.12m/s and the maximum rotational velocity was 8 degrees/s. Observation updates were

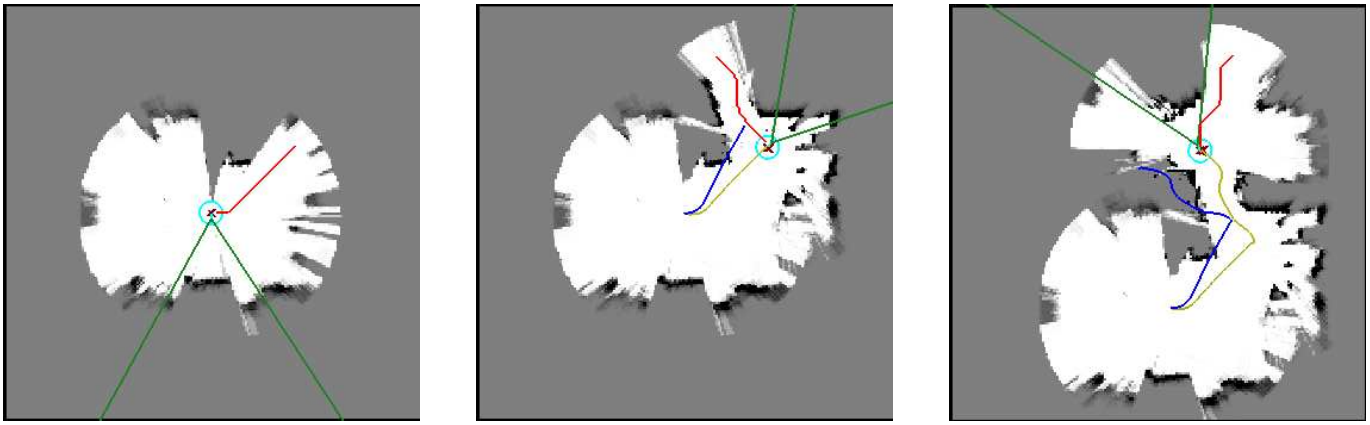

Fig. 4. Sample exploration sequence. Red: the planned route to the next goal; gold: the robot's trajectory and blue: the odometer.

Fig. 5. Sample images of the environment. Challenges include saturation from natural light, illumination variation, untextured surfaces, and obstacles that are too tall, or too short for detection with a laser.

triggered by robot motion estimates exceeding 7cm in translation or 1 degree in rotation, resulting in 2209 observation updates, at approximately 1.1Hz. The RBPF used 1000 samples and can easily manage up to 2000 samples on a dual 3.2GHz Pentium Xeon computer. We have found that the main cost of processing frames is the SIFT computation.

The robot traveled an estimated distance of 120m, and mapped approximately 7278 landmarks in about 35 minutes.

Figure 6 depicts the occupancy grid constructed from the robot's exploration. The grid resolution is 0.01m by 0.01m per pixel and it accurately captures the topology of the environment. Many of the cluttered regions correspond to office chairs and other furniture. Figure 7 depicts the landmark map for landmarks that were observed more than 3 times and which were not considered to belong to the floor or ceiling. While the landmark map appears cluttered, the accuracy of the occupancy grid demonstrates that the landmark map provided SLAM estimates that were accurate enough to maintain a good map.

Figure 8 depicts the difference between the filter's estimated trajectory (shown in gold), and the robot's odometer (shown in blue). Clearly the filter estimate is correct and consistent in that it successfully navigates through the door on several occasions, and it successfully locates the second, narrower door on the left side[2]. There is some minor deviation in the orientation of the two rooms which may pose difficulties in larger environments. The figure also depicts the robot's planned trajectory (shown in red) to an exploration goal in the lower left. For a comparison with raw odometry, Figure 9 depicts the deviance between the robot's odometer and the filter estimate over time. These results clearly demonstrate

[2]This second door is too narrow for the PowerBot to navigate safely.

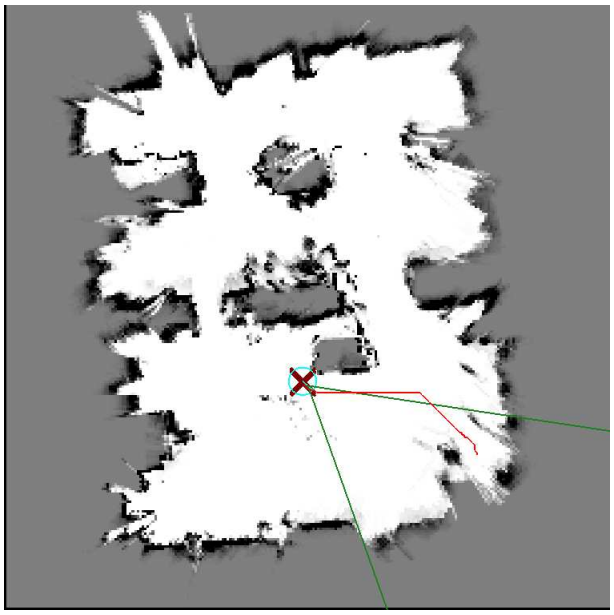

Fig. 6. Occupancy grid constructed for the maximum-likelihood sample at the end of exploration.

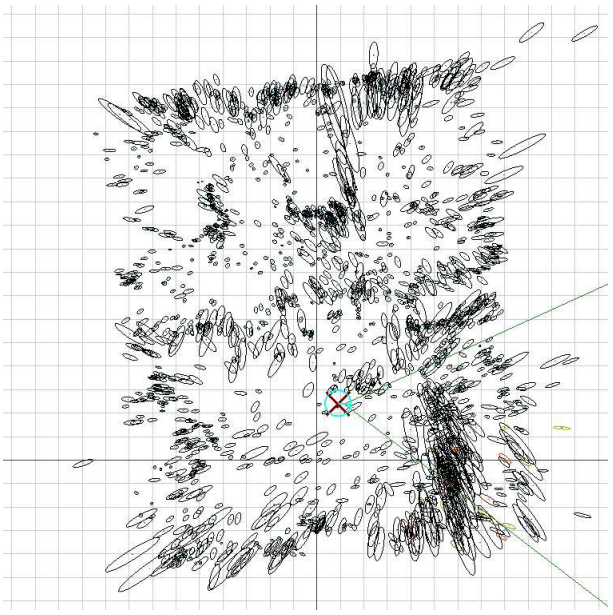

Fig. 7. Landmark map constructed for the maximum-likelihood sample at the end of exploration.

how unreliable the robot's differential drive and odometry is for estimating motion.

VIII. Conclusion

This paper has demonstrated a system that is capable of mapping a large, complex visual environment in real time. The system explores and navigates fully autonomously and can generate a hybrid map representation that facilitates accurate localization (using visual landmarks) and safe, robust navigation (using occupancy grids). Our system models SIFT features for landmarks, and uses trinocular stereo to produce occupancy grids. The topological accuracy of the occupancy maps are due to the localization accuracy of the underlying landmark-driven Rao-Blackwellised particle filter.

While our current work has demonstrated that robust autonomous vision-based mapping can be accomplished, there are several outstanding issues to pursue. First, a more thorough reliability study is required to understand the conditions that ensure filter convergence and to direct exploration in ways that prevent filter divergence. Second, we hope to implement gaze planning (swiveling the camera on a pan-tilt unit) for more reliable obstacle detection and avoidance, and add more sophisticated exploration strategies for exploring larger environments. We are also interested in applying more sophisticated proposal distributions [7] and running experiments in outdoor environments.

Acknowledgements

The authors gratefully acknowledge the invaluable assistance of Pantelis Elinas in the development of this work.

References

[1] C. Stachniss, G. Grisetti, and W. Burgard, "Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Barcelona, Spain, 2005, pp. 667–672.

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

[3] 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.

[4] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE. Peurto Rico: IEEE Press, June 1997, pp. 1000–1006.

[5] 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, pp. 593–598.

[6] R. Sim, M. Griffin, A. Shyr, and J. J. Little, "Scalable real-time vision-based SLAM for planetary rovers," in *IEEE IROS Workshop on Robot Vision for Space Applications*, IEEE. Edmonton, AB: IEEE Press, August 2005, pp. 16–21.

[7] P. Elinas, R. Sim, and J. J. Little, "σSLAM: Stereo vision SLAM using the Rao-Blackwellised particle filter and a novel mixture proposal distribution," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE. Orlando, FL: IEEE Press, May 2006, to appear.

[8] B. Yamauchi, A. C. Schultz, and W. Adams, "Mobile robot exploration and map-building with continuous localization," in *IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 16-21 1998, pp. 2833–2839.

[9] D. Murray and J. J. Little, "Using real-time stereo vision for mobile robot navigation." *Autonomous Robots*, vol. 8, no. 2, pp. 161–171, 2000.

[10] 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.

[11] D. Hähnel, D. Fox, W. Burgard, and S. Thrun, "A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements," in *Proc. of the Conference on Intelligent Robots and Systems (IROS)*, 2003.

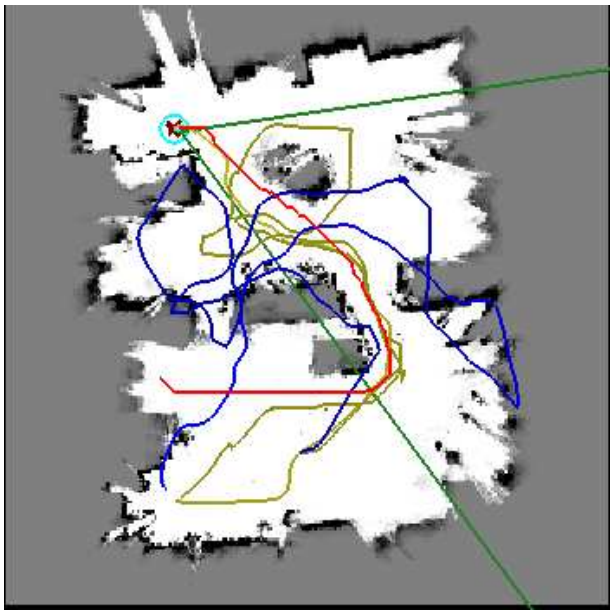

Fig. 8. Filter trajectory (gold) versus odometric trajectory (blue), and the robot's exploration plan (red).

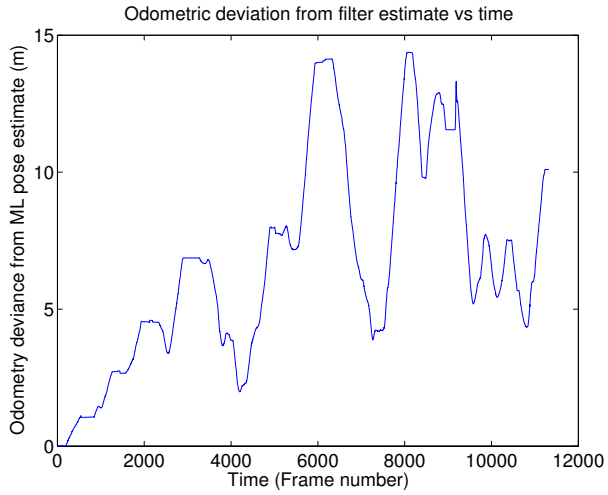

Fig. 9. Odometry deviance from filter estimate versus time.

[12] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[13] 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.

[14] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005, pp. 2428–2435.

[15] A. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings of the IEEE Int. Conf. on Computer Vision*, Nice, France, 2003, pp. 1403–1410.

[16] 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.

[17] 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.

[18] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE. Barcelona, Spain: IEEE Press, April 2005, pp. 673–678.

[19] S. Huang, N. M. Kwok, G. Dissanayake, Q. P. Ha, and G. Fang, "Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity," in *Proc. Int. Conf. on Robotics and Automation*, IEEE. Barcelona, Spain: IEEE Press, April 2005, pp. 1103–1108.

[20] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003.

[21] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Proc. of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2005, pp. 65–72.