

Effective Exploration Strategies for the Construction of Visual Maps

Robert Sim and Gregory Dudek
Centre for Intelligent Machines, McGill University
3480 University St, Montréal, Québec, Canada H3A 2A7
{dudek,simra}@cim.mcgill.ca

Abstract— We consider the effect of exploration policy in the context of the autonomous construction of a visual map of an unknown environment. Like other concurrent mapping and localization (CML) tasks, odometric uncertainty poses the problem of introducing distortions into the map which are difficult to correct without costly on-line or post-processing algorithms. Our problem is further compounded by the implicit nature of the visual map representation, which is designed to accommodate a wide variety of visual phenomena without assuming a particular imaging platform, thereby precluding the inference of scene geometry. Such a representation presents a requirement for a relatively dense sampling of observations of the environment in order to produce reliable models. Our goal is to develop an on-line policy for exploring an unknown environment which minimizes map distortion while maximizing coverage. We do not depend on costly post-hoc expectation maximization approaches to improve the output, but rather employ extended Kalman filter (EKF) methods to localize each observation once, and rely on the exploration policy to ensure that sufficient information is available to localize the successive observations. We present an experimental analysis of a variety of exploratory policies, in both simulated and real environments, and demonstrate that with an effective policy an accurate map can be constructed.

I. INTRODUCTION

In this paper we consider the problem of automatically exploring an environment and constructing a visual map. In particular, we are interested in selecting an exploration strategy which minimizes map uncertainty on-line. Such uncertainty is accumulated by errors in odometric sensing, and can grow unbounded over time. Our work differs from other exploration techniques in that the map representation is implicit in nature; that is, we do not produce a geometric description of the environment but rather a mapping from image features to robot pose. As such, it precludes explicit geometric inference of landmark positions and hence is not immediately suited to standard Kalman filter, or expectation maximization techniques [1], [2], [3]. Rather, our goal is to minimize error by selecting an appropriate exploration trajectory which allows the robot to localize against its known map as accurately as possible prior to adding a new observation to the map. We examine a variety of such trajectories, both in simulation and using a real robot, and present experimental results.

Our approach is aimed at using a purely on-line exploration paradigm to produce a map that is suitable for robotic navigation. Of course, in many contexts it might be desirable to post-process such a map to further optimize its accuracy, but the current work is motivated by the supposition that even in such cases a good initial map is helpful.

A key component of our work is the visual map representation [4]. Unlike the vast majority of mapping paradigms, which employ range sensors derived from sonar, laser or stereo cameras, visual maps make no attempt to infer scene geometry, but rather encode visual landmarks implicitly in the image domain. As such, visual maps do not require camera calibration, and can encode a wide variety of visual phenomena, including exotic phenomena such as specularities, shadowing and atmospheric hazing, using arbitrary imaging geometry. The challenge posed by such a representation is that unlike geometric landmarks it involves no explicit parameters to estimate, and hence is not well-suited to filters which aim to compute maximum-likelihood parameterizations. This poses an interesting challenge for autonomous mapping—how to maximize the map likelihood without an explicit representation. It should also be noted that, by definition, an implicit representation makes no prior assumptions about imaging geometry, so a further challenge is to infer the correct map without effectively linearizing away the nonlinear interactive properties of the environment and sensor. Finally, an implicit representation also poses the requirement for a relatively dense sampling of observations covering the pose space. This presents additional challenges in that the exploratory trajectory can be quite long, even in a small environment.

With these challenges in mind, our goal is to develop a technique for maximizing coverage of a relatively small pose space in order to generate an accurate visual map. The mapping process can be made more robust by composing a large map using a set of smaller sub-maps [5], [6], [7], [8]. These principles should be applicable to the mapping context described here, but can be viewed as a secondary stage of processing and control.

In the following sections we examine previous work

on the CML problem, followed by a description of the visual mapping framework. We then go on to establish a framework for exploration and discuss several candidate exploration policies. Finally, we present experimental results based on both simulation and validation in our lab, and discuss their implications.

II. RELATED WORK

The problem of concurrent mapping and localization (CML), also known as simultaneous localization and mapping (SLAM) has received considerable attention in the robotics community [9], [10], [3], [11], [12]. The state of the art in CML can be broadly subdivided into one of two approaches (and various hybrids). One family of methods collects measurements and incrementally builds the map while the robot moves (i.e. in an on-line fashion). Usually the map is represented as a set of landmarks derived from a range sensor, and a Kalman filter is employed to minimize the total uncertainty of the robot pose and the individual landmark positions [1], [2]. These techniques differ from previous Kalman filters employed for localization ([9], [10]) in that the landmark positions, as well as the robot pose, are being estimated. While there exist approximation techniques for reducing the computational expense of on-line CML (c.f. [13]), each update in the standard on-line approach is quadratic in the number of landmarks.

The second family of methods for CML involves first collecting measurements and then post-processing them in a batch. The standard post-processing method is to employ Expectation Maximization (EM), again to minimize the total uncertainty of robot poses and landmark positions [3]. One goal of our work is to develop an on-line exploration method which maximizes the accuracy of the map without resort to expensive map updating. While outside the scope of this paper, this result can in turn be employed as a reliable prior for subsequent EM-style post-processing.

While most of the prior work on mobile robot mapping exploits the use of range data to construct an explicit geometric map, several authors have considered the use of visual data. Nayar, *et al* were among the first to consider the use of a purely appearance-based representation of the world for robot navigation using principal components analysis (PCA) [14]. Pourraz and Crowley considered the stability of PCA-based methods for appearance-based navigation of a mobile robot [15] and Jugessur and Dudek looked at voting-based methods to make PCA methods robust to changes in the scene or illumination [16]. Several authors have also considered the use of vision-based sensing to extract a geometric map, which can then be used in a more traditional CML context. Se, *et al* extract stereo-based landmarks using a scale-invariant filter [17], and Davison and Kita considered the problem of actively servoing a stereo head for landmark acquisition as a robot traverses uneven terrain [18]. Finally, Dellaert *et al* take

advantage of environmental invariants, such as a planar ceiling, to construct a mosaic-like map by registering an ensemble of images [19].

III. VISUAL MAPS

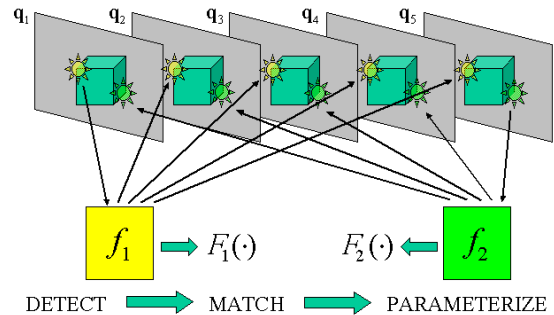


Fig. 1. Landmark Learning Framework: Salient features are detected in the input images and tracked across the ensemble. The resulting feature sets are subsequently parameterized as functions of the robot pose.

Our visual map representation employs the landmark learning framework described by Sim and Dudek [4]. We review it here in brief and refer the reader to the cited work for further details.

The key idea is to learn a set of visual features of a scene, and parametrically describe them so that they can be used to estimate one's position (that is, they can be used for localization). The features are pre-screened using an attention operator that efficiently detects statistically anomalous parts of an image and robust, useful features are tracked over an image ensemble and recorded along with an estimate of their individual utility.

The framework operates as follows. Assume for the moment that we have collected an set of observations of a scene with ground-truth position information associated with each image. The landmark learning framework operates by first selecting a set of local features from the images using a measure of visual attention, tracking those features across the ensemble of images by maximizing the correlation of the local image intensity of the feature, and subsequently parameterizing the set of observed features in terms of their behaviour as a function of the known positions of the robot (Figure 1). In the context of mapping, as each image arrives, matches to the parameterized features are located in the image, the image is localized from the matches using a Kalman Filter (see below), and the matched observations are inserted into the map using the filter estimate as the observation pose.

The feature parameterization employed in [4] computes a radial basis network interpolator of feature properties such as local appearance and image position. In practice, an arbitrary interpolation scheme can be employed and in

this work, for reasons of efficiency, we approximate the interpolant using bilinear interpolation over the Delaunay triangulation of the observation poses. Furthermore, again for efficiency reasons we measure only the image position of the landmark. Our previous work indicates that while the local appearance distribution is informative for localization, image position is a stronger and more compact indicator of pose [20].

IV. EXPLORATION FRAMEWORK

We have adapted the Extended Kalman Filter (EKF) localization framework described in the seminal papers by Smith *et al* and Leonard and Durrant-Whyte as the basis for our exploration framework [9], [10]. While the work by Leonard *et al* employed geometric beacons derived from range sensors as landmarks, the visual map representation instead employs landmark observations in the visual domain. It should be noted that, unlike EKF implementations deployed for CML which encode both robot pose and landmark position parameters, the only parameters maintained in our implementation are those of the robot pose. Given that the EKF has been studied extensively, we repeat here only those aspects of our implementation that are particular to our work.

At each time step k , the robot executes an action $\mathbf{u}(k)$, and takes a subsequent observation \mathbf{z} . The plant model is updated from \mathbf{u} according to the standard EKF formulation, and a set of matches to known landmarks \mathbf{z}_i are extracted from the observed image. Given that the visual map assumes a 2D configuration space, (that is, fixed orientation) some rehearsal procedure may be required to align the camera prior to taking an observation— we consider this issue in further detail in presenting our experimental results.

For each successfully matched landmark, a predicted observation $\hat{\mathbf{z}}_i$ is generated using the visual map, and the innovation $v_i(k+1)$ is computed

$$v_i(k+1) = \mathbf{z}_i(k+1) - \hat{\mathbf{z}}_i(k+1) \quad (1)$$

The innovation covariance requires estimation of the Jacobian of the predicted observation given the map and the plant estimate. We approximate this Jacobian as the gradient of the nearest face of the model triangulation and define it as $\nabla \mathbf{h}_i$. Defined as such, the innovation covariance follows the standard observation model:

$$\mathbf{S}_i(k+1) = \nabla \mathbf{h}_i \mathbf{P}(k+1|k) \nabla \mathbf{h}_i^T + \mathbf{R}_i(k+1) \quad (2)$$

where \mathbf{P} is the pose covariance following the action \mathbf{u} , and \mathbf{R} is the cross-validation covariance associated with the learned landmark model. It is important to note that \mathbf{R} serves several purposes— it is simultaneously an overall indicator of the quality of the interpolation model, as well as the reliability of the matching phase that led to the observations that define the model; finally it also accommodates the stochastic nature of the sensor.

A. Outlier Detection

It should be noted that feature correspondence takes place once an observation is obtained. However, there may be outlier matches that must be filtered out. As such, we employ the gating procedure described in [10], with the additional constraint that the gating parameter g is computed adaptively. Specifically, we accept landmark observations that meet the constraint

$$v_i(k+1) \mathbf{S}_i^{-1}(k+1) v_i^T(k+1) \leq g^2 \quad (3)$$

where

$$g = \max(g_{base}, \bar{g} + 2\sigma_g) \quad (4)$$

and g_{base} is a user defined threshold, and \bar{g} and σ_g are the average and standard deviation of the set of gating values computed for each landmark observation (that is, the left-hand side of Equation 3). This selection of g allows the filter to correct itself when several observations indicate strong divergence from the predicted observations— indicating a high probability that the filter has diverged and affording the opportunity to correct the error.

B. Map Update

Given the set of gated observations, the EKF is updated according to the standard formulation, whereby the set of filtered innovation measurements is compounded into a single observation vector and a least-squares solution is computed for the Kalman gain. Combined with the plant model, a pose estimate and associated covariance are obtained. Once an updated pose estimate is available, the successfully matched landmarks are inserted into the visual map, using the estimated pose as their observation pose. It should be noted that we also insert those observations that were removed by the gating procedure. We take this approach because it serves to increase the cross-validation covariance associated with the mis-matched landmark, thereby reducing its influence for future localization. As such, at the end of the exploration procedure, only those landmarks that serve to match reliably *and* localize reliably can be selected and retained.

In the subsequent section we consider the problem of selecting exploration trajectories that result in an accurate map using the EKF framework.

V. EXPLORATION POLICIES

We are interested in comparing candidate robot exploration policies with the goal of balancing two competing interests: coverage and accuracy. In other words, we want to build the largest, most accurate map possible in a finite amount of time. Given that there are an infinite number of possible exploration trajectories, we will restrict our consideration to a set of policies which are either intuitively satisfying or serve to illustrate an extreme case. The particular policies we will examine are described below. They

are *SeedSpreader*, *Concentric*, *FigureEight*, *Random*, *Triangle* and *Star*. An example of each trajectory is depicted in Figure 2.

a) SeedSpreader: The robot follows a seed-spreader pattern through the environment [21]. We employ a variation on this approach by oscillating in a zig-zag motion as the robot moves along the path. This is performed in order to ensure that the visual map spans two dimensions, even along the first pass of the seed spreader.

b) Concentric: The robot traces a series of concentric circles out from its starting point, reversing direction for alternating circles.

c) FigureEight: Like the *Concentric* pattern, the robot traces a set of concentric circles, but in a series of growing figure-eights, bringing the robot close to its starting point with each pass.

d) Random: The robot takes a random step in a random direction at each time step.

e) Triangle: The robot traces a series of concentric equilateral triangles, taking observations at fixed intervals along each side. The advantage of this approach is that the ideal set of observation poses covers the pose space in a uniform tiling.

f) Star: The robot oscillates along a set of rays emanating from its starting point. The rays grow in length over time such that the set of observation poses is roughly the same as that for *Concentric*.

In all cases, the exploration model follows a *Plan, Act, Observe, Update* loop, planning a motion based on the exploration policy and the covered trajectory, executing the action, taking an observation, and updating the Kalman Filter and visual map. A single action is either a rotation or a translation, and new images are obtained only after a translation. Furthermore, while the filter is always updated after obtaining a new image, the image is only added to the visual map if there are no previous images from nearby poses in the map. As such, when the robot traverses previously explored territory, it localizes without updating the map.

A. Safety

In all robotic applications, special attention must be devoted to the safety of the robot and other agents in the environment. Given that the visual map does not encode geometric information, obstacle inference and avoidance requires careful consideration. We employ a sonar sensor as a virtual bumper, using the last sonar observation to determine whether an action is safe or unsafe to execute.

VI. EXPERIMENTAL RESULTS

In this section we present the results of our experiments in simulation and with a real robot. Our approach involves executing each exploration policy in turn and computing the deviation between the estimated trajectory and the actual trajectory of the robot.

A. Simulation

Our experimental setup for the simulated experiments is as follows: a simulated robot is placed in one room of a 1200cm by 600cm rectilinear environment. The robot has a ring of sixteen evenly spaced sonar sensors for detecting collisions, and the robot’s odometry model is set to add normally distributed zero-mean, 1% standard deviation error to translations and normally distributed zero-mean, 2% standard deviation error to rotations. The camera model is simulated by texture-mapping the walls with real images of our laboratory, and rendering the scene based on the ground-truth pose of the robot and a model of a simple perspective camera. As we have previously mentioned, we assume that the camera has the ability to align itself using a procedure which is external to the robot drive mechanism, possibly using a compass and pan-tilt unit or an independent turret, such as that which is available on a Nomad 200 robot. Using this procedure, when an image observation is required, the camera snaps two images, one along the global x axis and one along the global y axis, and returns the composite image. Figure 3 illustrates a typical image returned by the camera in one direction in the simulated environment.

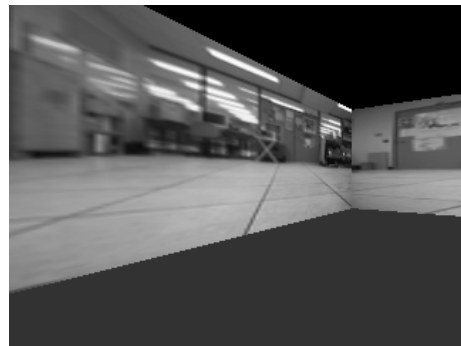


Fig. 3. Simulated camera view.

Each exploratory policy was run in the simulated environment, executing actions and taking observations until one of two conditions was met: either the visual map contained two hundred images, or the robot was unable to execute its action safely. The starting pose of the robot was selected to be the center of the room, except in the case of the *SeedSpreader*, which started in the corner.

The results of the experiments are tabulated in Table I. For each policy we report the mean deviation between the filter pose estimate and ground truth and the mean deviation between odometry and ground truth. We also record an estimate of the space explored per unit time (exploration efficiency), expressed as the total number of observation images inserted into the visual map divided by the total number of actions executed by the robot. A small value indicates that the robot spent most of its time in

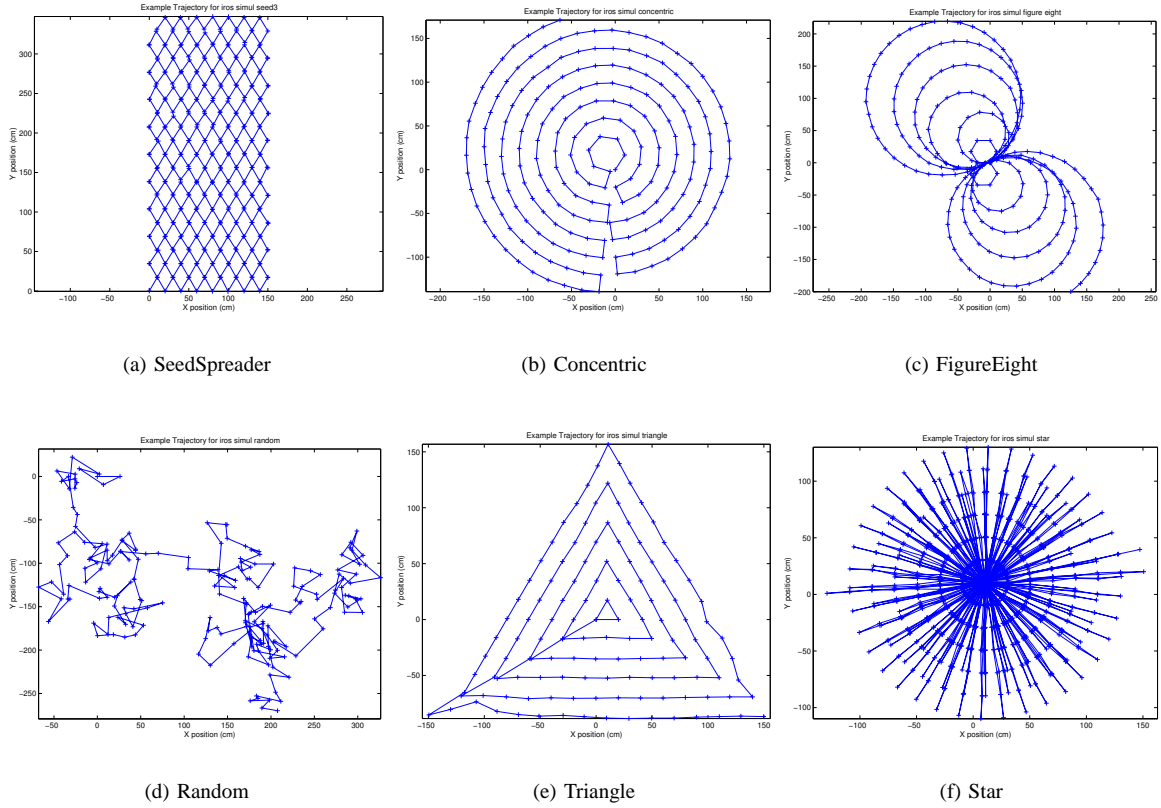


Fig. 2. Example Trajectories by Policy

TABLE I
SUMMARY OF EXPLORATION RESULTS BY EXPLORATION POLICY.

Method	Mean Filter Error (cm)	Mean Odometric Error (cm)	Exploration Efficiency (images/step)	Maximal Distance from O (cm)
SeedSpreader	26.4	43.7	0.178	373
Concentric	9.57	4.95	0.496	183
FigureEight	5.09	6.87	0.411	242.9
Random	8.46	93.4	0.475	347
Triangle	30.1	12.0	0.480	173
Star	1.63	23.8	<0.001	152

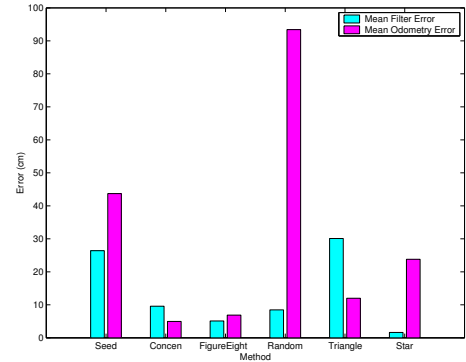


Fig. 4. Mean Filter Error and Odometry Error for each method

previously explored space. Finally, we report the maximal distance achieved from the robot’s starting pose.

Figure 4 summarizes the mean filter error and mean odometry error for each method and Figure 5 summarizes the exploration efficiency, expressed as the number of images inserted into the map per robot action. Note that while most of the methods inserted images at a near-optimal rate (after each rotate-translate pair of actions), as expected, the Star policy is highly inefficient as it repeatedly traverses previously explored terrain.

Figure 6 depicts propagation of error versus ground truth for the filter and odometer for each policy, sampled

at intervals of ten time steps. It is clear from these results that the Star policy produced the most accurate map, and the Random policy performed very well relative to the accumulated error. While the good performance of the Random policy is likely due to the fact that it occasionally re-traverses old territory, it is no doubt an unsuitable choice for task-driven robotics. It is perhaps surprising that the Concentric and FigureEight policies do not perform more accurately. The principal

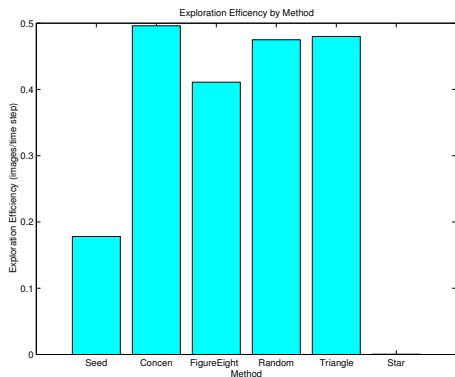


Fig. 5. Exploration Efficiency for each method.

cause is that as the circles get wider, errors from the inner circles are amplified by the linearization of the filter, and add bias to the localization estimates. In addition, the errors become highly correlated between successive observations, whereas such correlations are avoided using the *Star* method by relocalizing with respect to a reliable reference point (the centre of the star). Finally, note the sudden and extreme divergence of the *Triangle* method, a result of divergence in the filter as the robot failed to correctly turn the corner of one of the points of the triangle.

B. Real world performance

We have implemented the exploration framework on a Nomadics Super Scout mobile robot. The Scout platform employs a differential drive mechanism for steering and is particularly prone to rotation errors. The robot was equipped with a monocular camera and a KVH C100 compass. The compass was employed to align the camera—while local variations in magnetic field made the compass useless for navigation, the variations were repeatable as a function of pose and degraded smoothly enough that the robot could be steered to face in the direction of a particular heading when an image was captured. Nonetheless, some noise was observed in observation orientation, and this noise presented itself in relatively large cross-validation estimates for the x image position of any given landmark. Given the time required to align the camera, only one direction was imaged for each observation. The robot commenced exploration from the centre of an open space in our lab (Figure 7a). A sample image from the robot’s camera is shown in Figure 7b).

We ran two experiments on the robot, employing the *Concentric* and *Star* exploration policies respectively. Exploration continued until 100 images were inserted into the map. Since the ground truth trajectory was not available, when exploration terminated the robot was instructed to navigate back to its starting position. The discrepancy between the final position and the starting



(a) The robot in the environment.



(b) Robot’s eye view of the scene

Fig. 7. The real-world environment and robot’s eye view.

TABLE II
FINAL POSE ERRORS BY EXPLORATION POLICY FOR THE REAL ROBOT.

Method	Actual Final Pose (x,y)(cm)	Odometer Pose (x,y)(cm)	Control Pose (x,y)(cm)
Concentric	(6,-14)	(55,20)	(-6,-15)
Star	(21,2)	(171,133)	(24,-23)

position was measured by hand. Figure 8 depicts the filter trajectory for each method.

The discrepancy between the robot’s starting and ending positions are shown in Table II. In all cases, the robot started at pose $(0,0)$, and ended the trajectory by homing until the filter indicated a pose within 3cm of the origin. Depicted in the table are the actual ending pose (measured by hand), the ending pose reported by the odometer, and the ending pose reported by the control inputs. All measurements are in centimetres. The filter outperformed both odometry and the control inputs in both cases, albeit *Concentric* demonstrated only a marginal improvement over the control estimate. It is remarkable

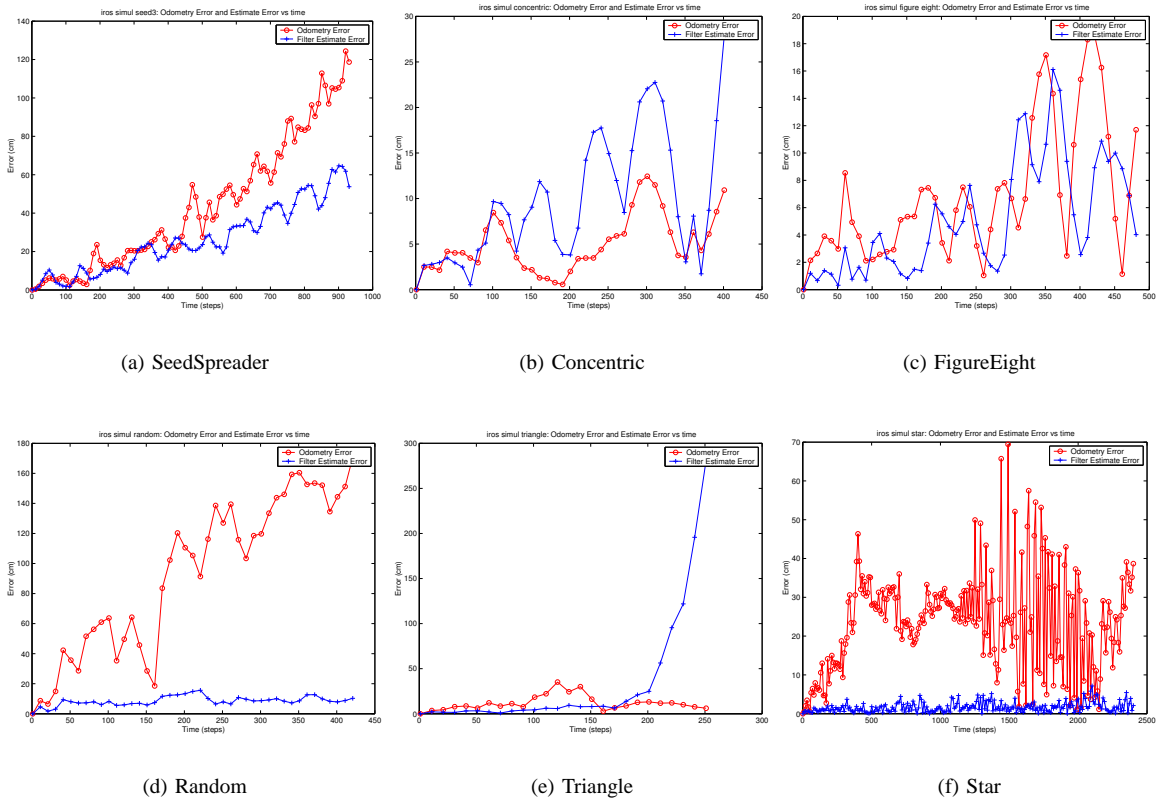


Fig. 6. Filter and Odometry Error plotted versus Time for each Method

to note the degree to which the odometer diverged even from the control inputs.

VII. DISCUSSION AND CONCLUSIONS

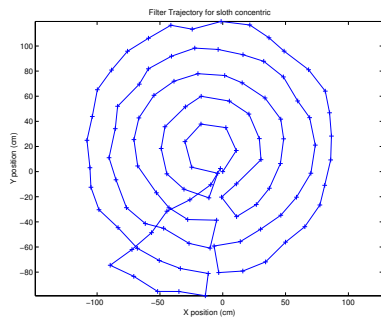
In this paper we have examined the problem of automatically constructing a visual map of the environment, with particular attention paid to selecting an exploration policy that balances accuracy with efficiency. Our simulated results indicate that this particular balance is difficult to strike, as the best way to improve accuracy is to select a highly inefficient method. We also presented results using a real robot and, while the lack of ground truth presents a difficulty in evaluating the results, the homing strategy indicated that the resulting map was useful for navigation and far more accurate than odometry-based navigation.

One interesting venue for future work is the combination of appearance-based mapping, as presented here, with simultaneous geometric mapping and localization using range estimates extracted from the vision sensor (and/or from supplementary data sources). While the appearance-based method has certain advantages in terms of its freedom from strong dependence on assumptions about scene geometry, reflectance, lighting, etc., it does have shortcomings that might be addressed by the use of geo-

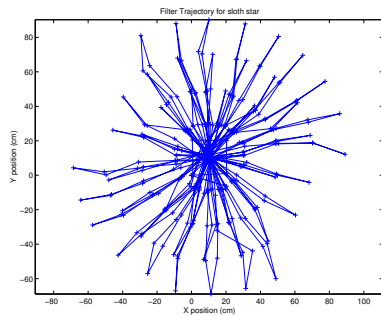
metric methods whose failure modes are complementary to those of the appearance-based technique.

VIII. REFERENCES

- [1] J. J. Leonard and H. J. S. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *Robotics Research: The Ninth International Symposium* (J. Hollerbach and D. Koditschek, eds.), (London), Springer-Verlag, 2000.
- [2] J. Guivant, E. Nebot, and H. Durrant-Whyte, "Simultaneous localization and map building using natural features in outdoor environments," in *Sixth Int. Conf. on Intelligent Autonomous Systems*, (Italy), 2000.
- [3] S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Machine Learning*, vol. 31, pp. 29–53, 1998.
- [4] R. Sim and G. Dudek, "Learning generative models of scene features," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, (Lihue, HI), IEEE Press, December 2001.
- [5] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of



(a) Concentric



(b) Star

Fig. 8. Filter trajectory for each method

spatial representations,” *Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.

- [6] S. Simhon and G. Dudek, “A global topological map formed by local metric maps,” in *IEEE/RSJ Int. Conf. on Intelligent Robotic Systems*, (Victoria, Canada), October 1998.
- [7] I. Ulrich and I. Nourbakhsh, “Appearance-based place recognition for topological localization,” in *Proc. IEEE Intl. Conf. on Robotics and Automation*, pp. 1023–1029, IEEE Press, April 2000.
- [8] H. Choset and K. Nagatani, “Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization,” *IEEE Trans. on Robotics and Automation*, vol. 17, April 2001.
- [9] R. Smith, M. Self, and P. Cheeseman, “Estimating uncertain spatial relationships in robotics,” in *Autonomous Robot Vehicles* (I. Cox and G. T. Wilfong, eds.), pp. 167–193, Springer-Verlag, 1990.
- [10] J. Leonard and H. F. Durrant-Whyte, “Simultaneous map building and localization for an autonomous mobile robot,” in *IEEE Int. Workshop on Intelligent Robots and Systems*, (Osaka, Japan), pp. 1442–1447, November 1991.

- [11] B. Yamauchi, A. Schultz, and W. Adams, “Mobile robot exploration and map building with continuous localization,” in *IEEE Int. Conf. on Robotics and Automation*, (Leuven, Belgium), pp. 3715–2720, IEEE Press, May 1998.
- [12] H. Blaasvaer, P. Pirjanian, and H. I. Christensen, “Amor: An autonomous mobile robot navigation system,” in *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 2266–2271, 1994.
- [13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *AAAI National Conference on Artificial Intelligence*, (Edmonton, Canada), AAAI, 2002.
- [14] S. Nayar, H. Murase, and S. Nene, “Learning, positioning, and tracking visual appearance,” in *IEEE Conf on Robotics and Automation*, (San Diego, CA), pp. 3237–3246, May 1994.
- [15] F. Pourraz and J. L. Crowley, “Continuity properties of the appearance manifold for mobile robot position estimation,” in *2nd IEEE Workshop on Perception for Mobile Agents*, (Ft. Collins, CO), IEEE Press, June 1999.
- [16] G. Dudek and D. Jugessur, “Robust place recognition using local appearance based methods,” in *IEEE Conf on Robotics and Automation*, (San Francisco), IEEE Press, April 2000.
- [17] S. Se, D. Lowe, and J. Little, “Vision-based mobile robot localization and mapping using scale-invariant features,” in *IEEE Conf on Robotics and Automation*, (Seoul, Korea), pp. 2051–2058, May 2001.
- [18] A. J. Davison and N. Kita, “Sequential localisation and map-building for real-time computer vision and robotics,” *Robotics and Autonomous Systems*, vol. 36, no. 4, pp. 171–183, 2001.
- [19] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, “Using the condensation algorithm for robust, vision-based mobile robot localization,” in *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, June 1999.
- [20] R. Sim and G. Dudek, “Learning and evaluating visual features for pose estimation,” in *IEEE Int Conf on Computer Vision*, (Kerkyra, Greece), IEEE Press, sept 1999.
- [21] S. Lumelsky, S. Mukhopadhyay, and K. Sun, “Dynamic path planning in sensor-based terrain acquisition,” *IEEE Trans Robotics and Automation*, vol. 6, no. 4, pp. 462–472, 1990.