Invariant Filtering for Simultaneous Localization and Mapping

Matthew C. Deans

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, 15217, USA

Martial Hebert

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, 15217, USA

Abstract

This paper presents an algorithm for simultaneous localization and map building for a mobile robot moving in an unknown environment. The robot can measure only the bearings to identifiable targets and its own relative motion. The approach is to recursively estimate features of the environment which are invariant to the robot pose in order to decouple the pose error from the map error. The highly nonlinear nature of this problem requires more explicit reasoning about the spatial relationships between landmarks and between the robot and landmarks than those used in previous methods.

1 Introduction

It may be required for a robot to enter an unknown environment and to concurrently explore and produce a metric map of the area while maintaining an accurate estimate of its position. If the robot were to have an *a priori* map, then localization would be a relatively easy task. Alternatively, if the robot were to have a precise, externally referenced position estimate, then mapping would be an easy task. However, problems in which the robot has no *a priori* map and no external position reference are particularly challenging. Such scenarios may arrise for AUV's, mining applications, or planetary surfaces. This problem has been referred to as *concurrent localization and mapping* (CLM) and *simultaneous localization and mapping* (SLAM). We will use the latter in this paper.

Section 2 discusses some previous work in SLAM. Section 3 discusses the general ideas behind how the Invariant Filter works. Section 4 details the algorithm. Section 5 shows results from implementing the filter and testing it using real data.

2 Previous Work

The SLAM problem was originally taken on by Smith, Self and Cheeseman[1] who first published the *stochastic map*, and by Moutarlier and Chatila[2] who provided experiments using this approach with real data. Because the stochastic mapping approach is based on the Extended Kalman Filter, the update requires an $O(N^3)$ matrix inversion, where N is the number of landmarks or environment states to be estimated. Approaches based on the stochastic map method are common and a good review can be found in [3].

Recently, however, new solutions to the SLAM problem have been reported which take a principled approach to decoupling the problem in order to keep the computation tractable. If the state covariance matrix is diagonal or block diagonal, inversion becomes simpler and the EKF update can be done much faster. Leonard and Feder have recently introduced the Decoupled Stochastic Mapping (DSM) algorithm[4] which tesselates the environment with loosely coupled local maps, each of which is a full stochastic map. Between local maps the relationship is approximate and covariances are not considered. The computational complexity of the *stochastic map* is incurred only within each local map.

Csorba and Durrant-Whyte introduced the concept of a *relative filter*[5], which recursively estimates features which are invariant to robot position, thereby removing the correlation between the robot position uncertainty and feature uncertainties, as well as feature-feature correlations. Newman expanded on the method, introducing the Geometric Projection Filter (GPF)[6] which provides a means by which to produce a geometrically consistent map from the relative features in Csorba's filter by solving a set of simultaneous linear constraint equations to project the relative map states onto an absolute coordinate system. To the authors' knowledge, previous implementations of

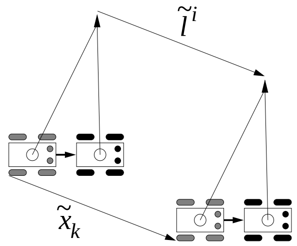

Figure 1: Errors in estimates of rover pose and landmark position, in the global frame, are correlated

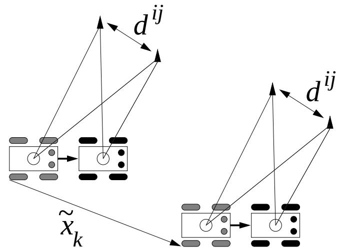

Figure 2: Environment features invariant to rover pose are impervious to errors in pose estimate

the relative filter and GPF have assumed that absolute heading is measurable (e.g. by a compass) and that bearing and range to each target is measurable. Under these conditions, the relative filter or GPF can use environment features which are invariant only under translation and treat the problem in a linear fashion.

In this paper we present an algorithm for Invariant Filtering, which is a more general version of the GPF[6]. The approach is based on the same principles as the relative filter and GPF, but is able to deal with unknown heading as well by filtering environment features which are invariant under rotation and translation, not just translation. We also consider the use of a bearings only sensor, using an omnidirectional camera to track landmarks as the robot moves.

3 Invariant Filter

Consider a mobile robot which can measure the bearings to landmarks, which we will define to be uniquely identifiable point features. The robot can measure its own motion through odometry or inertial sensing. We will restrict our discussion here to the planar case. The robot's pose in (x, y, θ) is parameterized by the vector x_k. There is some nonlinear stochastic process model which, given an initial position estimate $\hat{x}_k$ and a motion measurement u_k, predicts the final position of the robot $\hat{x}_{k+1}$

$$\hat{x}_{k+1} = \mathcal{F}(\hat{x}_k, u_k) + w_k \tag{1}$$

where w_k is taken to be zero mean Gaussian noise with covariance Q. The robot is equipped with a bearing sensor which makes a noisy measurement of the bearing to a landmark from the robot's current pose,

$$z_k^i = h(x_k, \ell^i) + \nu_k^i \tag{2}$$

where ν_k is zero mean Gaussian noise with covariance R. If the robot measures the bearing to a landmark from two unique positions x_k and x_{k+1}, then an estimate of the location of the landmark may be inferred through triangulation. Let $\mathcal{L}()$ be a function which takes as inputs the starting pose of the robot, the motion measurement, and the bearing to a landmark from the starting and ending pose, and computes the position of the landmark by triangulation,

$$\ell^i = \mathcal{L}(x_k, u_k, z_k^i, z_{k+1}^i). \tag{3}$$

In any realistic scenario, the pose of the robot is not known to absolute certainty. The estimate $\hat{x}_k$ of the robot pose has some associated error $\tilde{x}_k = \hat{x}_k - x_k$ and landmark position estimates $\hat{\ell}^i$ have error $\tilde{\ell}^i = \hat{\ell}^i - \ell^i$. One major difficulty with absolute coordinate frames in SLAM is that landmark position estimates $\hat{\ell}^i$ and the robot pose estimate $\hat{x}_k$ are correlated. Assuming for the moment that the bearing measurements z_k^i and z_{k+1}^i are perfect, the error in the landmark position due to the error in robot pose can be found by expanding $\tilde{\ell}^i$ then taking a Taylor expansion

$$\begin{aligned} \tilde{\ell}^i &= \hat{\ell}^i - \ell^i \\ &\quad \mathcal{L}(x_k + \tilde{x}_k, u_k, z_k^i, z_{k+1}^i) - \\ &\quad \mathcal{L}(x_k, u_k, z_k^i, z_{k+1}^i) \\ \tilde{\ell}^i &\approx (\nabla_{x_k}\mathcal{L})\tilde{x}_k + O((\tilde{x}_k)^2) \end{aligned} \tag{4}$$

Dropping all but the first order term,

$$E[\tilde{x}_k \tilde{\ell^i}_k] = E[\tilde{x}_k (\nabla_{x_k}\mathcal{L})\tilde{x}_k] \tag{5}$$

Figure 1 shows a diagram of how this happens. The vehicle measures the bearing to a target, moves, and measures the bearing to the target again. The position of the target is measured in a rover-centered coordinate frame with the initial robot position as its origin. The resulting estimate of the landmark location in absolute coordinates depends on where the vehicle was when it began its motion and the error in landmark position is directly correlated with the error in robot position.

A similar argument demonstrates how the error in the robot pose estimate becomes correlated to errors in the map when the rover attempts to localize relative to landmarks with erroneous locations. Over time, filtering methods such as Kalman filters which recursively update robot pose and landmark positions in an absolute coordinate frame can diverge.

In addition, absolute filtering approaches to solving the SLAM problem suffer the curse of dimensionality. The state space is of dimension N where in a planar environment, N is three parameters for the robot pose plus two parameters for every landmark. Kalman filters require $O(N^2)$ storage and $O(N^3)$ computation per step in order to keep track of the full covariance matrix, and as the robot explores larger and larger regions, the filter becomes intractable.

The fundamental importance of the relative filter[5] is that it recursively updates estimates of environmental features which are invariant to the robot pose. Consider the distance between two landmarks,

$$d^{ij} = ||\ell^i - \ell^j||^2. \tag{6}$$

The estimate $\hat{d}^{ij}$ and error $\tilde{d}^{ij}$ do not depend on x_k. If $\mathcal{D}(x_k, u_k, z_k^i, z_k^j, z_{k+1}^i, z_{k+1}^j)$ is the estimator for d^{ij}, then

$$\nabla_{x_k}\mathcal{D}(x_k, u_k, z_k^i, z_k^j, z_{k+1}^i, z_{k+1}^j) = 0 \tag{7}$$

and therefore

$$E[\tilde{x}_k \tilde{d}_k^{ij}] = 0. \tag{8}$$

Even an error in heading, which is notoriously problematic for dead reckoning, does not affect the estimate. Figure 2 shows a robot estimating the distance d^{ij} between landmarks i and j. The position of each landmark is wrong due to $\tilde{x}_k$ but the relationship between the landmarks is unaffected.

The two invariant features which can be estimated are pairwise distances, as mentioned above, and angles ψ^{ijk} subtended by a landmark pair ℓ^i, ℓ^k as measured from a third, ℓ^j, as in Figure 3. The endpoints which define a distance, or the triplet of points which define an angle, are called *control points*. For simplicity, we will deal with only distance measurements for the rest of this paper, but note that angles may be more difficult to use. Angles represent more nonlinear relationships and may present singularities. In addition, estimating angles requires the simultaneous observation of three landmarks instead of two.

4 The Algorithm

In this section we will detail the the current implementation of the algorithm.

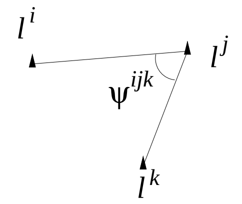

Figure 3: Angles between two landmarks subtended at a third are invariant to robot pose

4.1 Data structure

The invariant filter requires two separate "maps" to be maintained. The first map is a relative map similar to that of [5]. This map contains the invariant relationships between landmarks. Each invariant feature has an associated type (distance or angle), a list of the landmarks which make up its control points, and a mean and covariance for the estimated feature value.

The second map is an absolute map which contains estimates for landmark positions in a global coordinate frame. Each landmark in the absolute map contains an estimated (x, y) coordinate and a list of the invariant features for which it is a control point. This enables efficient computation of the derivatives for optimizing the absolute map with respect to the relative map. Figure 4 shows a diagram of these data structures and their relationships.

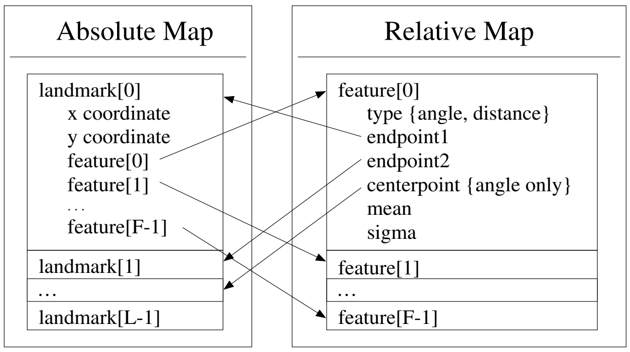

Figure 4: Relationship between absolute map and relative map data structures

4.2 Building the maps

When a new landmark is encountered, its position is estimated in the global reference frame using the estimate of the robot pose and an estimate of the landmark position in the robot coordinate frame. The landmark is then inserted into the absolute map at that location. It is only important that the landmark be inserted close enough to its "true" position that the optimization procedure used to update the absolute map converges to the right solution.

Invariant features involving the new landmark must also be inserted into the relative map. A greedy triangulation algorithm is used to connect the new landmark to observable neighbors. Note that we do not require a planar triangulation. The triangulation must simply provide enough constraints to fix the landmarks in the absolute map. An attempt is made to create invariant features using landmarks which are close together, since landmarks which are far apart are less likely to be observed simultaneously.

4.3 Strategy for avoiding correlations

The goal in this filtering method is to ensure that the state vector is decoupled so that memory requirements and update computation are trivial. In order to do this, we must be careful not to introduce correlations. Correlation between robot pose error and landmark position errors is eliminated by using map features which are invariant to robot pose. By proper consideration, we can also eliminate correlation between map features.

When bearing observations are made, the bearings $Z_k = \{z_k^i; i = 0, \ldots\}$ to all visible targets at time k are stored. The measurement u_k of motion from time k to $k+1$ is also stored.

Triangulation allows computation of the landmark positions using the motion estimate as a baseline,

$$\hat{\ell}^i = \mathcal{L}(x_k, \mathcal{F}(x_k, u_k), z_k^i, z_{k+1}^i). \tag{9}$$

We then compute the Euclidean distance between pairs of landmarks as $\hat{d}^{ij} = ||\hat{\ell}^i - \hat{\ell}^j||^2$. The error $\tilde{d}^{ij}$ will be correlated with u_k, z_k^i, z_{k+1}^i, z_k^j, and z_{k+1}^j.

In order to be sure that the estimate $\hat{d}^{ij}$ is not correlated with any other features in the relative map, we cannot use any of $\{u_k, z_k^i, z_{k+1}^i, z_k^j, z_{k+1}^j\}$ to estimate other features. The restriction that we cannot use z_{k+1}^i and z_{k+1}^j means that we must remove those measurements from the buffer. The restriction that we cannot use u_k means that not only will u_k be removed from the buffer, but also all old bearing measurements Z_k as well, since these bearings cannot be used for triangulation without estimates of motion after the bearings were measured. The result is that after step $k+1$, only $Z_{k+1} - \{z_{k+1}^i, z_{k+1}^j\}$ remain in the measurement buffer for the next step.

4.4 Updating the relative map

At time k, the algorithm checks the buffer to find all pairs of consecutive bearing measurements of the same landmark $\{z_{k-1}^i, z_k^j\}$. These pairs create a list of observable invariant features which are checked against the invariant features already in the relative map. This check is necessary because not every landmark pair distance or landmark triplet angle appear in the relative map.

Once a list of candidate features is found, a decision must be made as to which feature should be updated. We have adopted a greedy strategy where the observable relative feature $\hat{d}_k^{ij}$ with the highest covariance σ_k^{ij} is updated. The selected feature is updated using the familiar equations

$$\begin{aligned} \hat{d}_{k+1}^{ij} &= \frac{(\sigma_{\mathcal{D}})^2 \hat{d}_k^{ij} + (\sigma_k^{ij})^2 \mathcal{D}(\hat{\ell}_{k+1}^i, \hat{\ell}_{k+1}^j)}{(\sigma_{\mathcal{D}})^2 + (\sigma_k^{ij})^2} && (10) \\ \sigma_{k+1}^{ij} &= \frac{(\sigma_{\mathcal{D}})^2 (\sigma_k^{ij})^2}{(\sigma_{\mathcal{D}})^2 + (\sigma_k^{ij})^2} && (11) \end{aligned}$$

where $\hat{d}_k^{ij}$ and σ_k^{ij} are the estimate and variance at time k and $\mathcal{D}(\hat{\ell}_{k+1}^i, \hat{\ell}_{k+1}^j)$ is the new measurement, with variance $\sigma_{\mathcal{D}}$.

4.5 Updating the absolute map

The update of the absolute map is independent of the update of the relative map and can be done asynchronously. The upate step can be invoked automatically whenever the relative map is updated or can be delayed until a localization is required, a relative feature estimate changes dramatically, or some other criterion.

The update is done by optimizing the relationships between landmarks in the absolute map with respect to the relationships estimated in the relative map. This is done by applying Levenberg-Marquardt (LM) optimization to the cost function

$$E = \sum_{p=0}^{F_k} \left(\frac{\hat{d}^{i(p)j(p)} - \mathcal{D}(\hat{\ell}^{i(p)}, \hat{\ell}^{j(p)})}{\sigma^{i(p)j(p)}} \right)^2 \tag{12}$$

where F_k is the number of features in the relative map at time k and $i(p)$ and $j(p)$ are the indeces for the landmarks which are the control points for the p^{th} feature.

The LM algorithm has an advantage over the EKF or IEKF (i.e. Newton-Raphson) update in cases where inverse Hessian methods cause iterations to diverge away from the solution. The LM algorithm checks the solution at each step by evaluating (12) and if the solution is worse, LM smoothly transitions to gradient descent behavior by forcing the Hessian to be diagonally dominant[7]. One step of LM has the same complexity as one step of the EKF or IEKF.

4.6 Memory and computational requirements

At time k, let L_k be the number of landmarks in the absolute map, F_k be the number of features in the relative map and V_k be the number of landmarks that are visible. In the worst case, i.e. for small areas or long range sensors where every landmark is visible at each step, V_k could be L_k, but generally $V_k < L_k$. With a reasonable triangulation, F_k is $O(L_k)$.

Checking for new features to add is fast. A list is maintained containing all of the landmarks which require another invariant feature in order to be properly constrained. This list is usually much smaller than L_k. Determining whether a new landmark is to be added is $O(V_k)$, since all visible landmarks must be checked to see if any are new. Determining which invariant feature to update is $O(F_k)$ in the worst case. Once the feature to be added or updated is selected, the update is $O(1)$. $O(F_k)$ memory is required to store the relative map since a full covariance matrix is not needed.

The update of the absolute map given the features in the relative map is done using Levenberg-Marquardt optimization. One iteration is $O(L_k^3)$ in general, but may be faster if sparse matrix techniques are used to speed up the inversion of the Hessian. The algorithm has quadratic convergence near the optimum, so although we require an unknown number of iterations there should not be many if the solution is near the optimum when iteration begins. We also do not need to update the absolute map every cycle. $O(L)$ memory is required to store the absolute map.

By comparison, Kalman filtering requires an $O(L_k^3)$ update each time new observations become available, and $O(L_k^2)$ memory to maintain the covariance matrix.

5 Experimental Results

The above algorithm has been implemented and tested in simulation and on real data. In the interest of time we will only present results with real data here.

The robot is an RWI ATRV with an omnidirectional

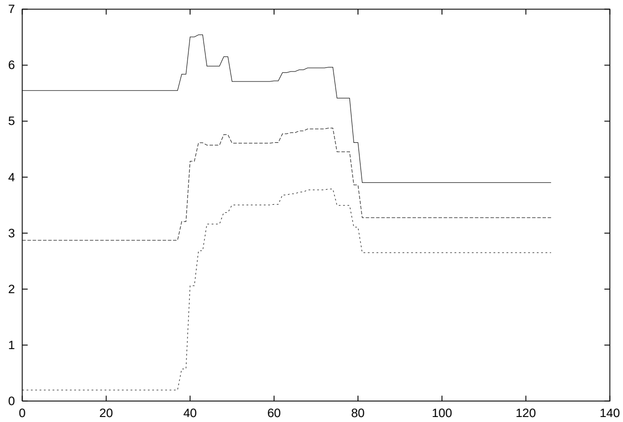

Figure 5: Convergence of relative feature 0, d^{01}.

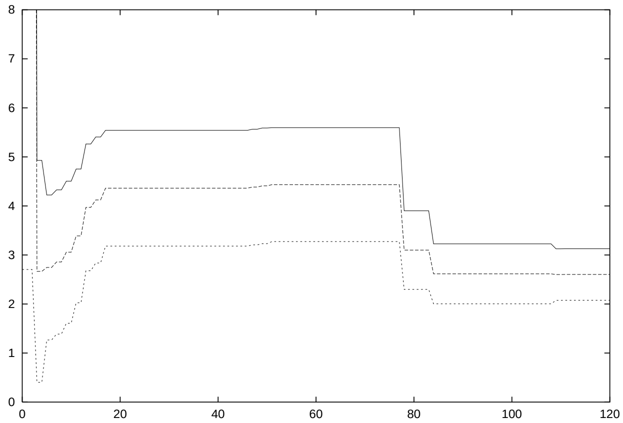

Figure 6: Convergence of invariant feature 4, d^{23}.

camera onboard. The robot logs images from the camera at about 1Hz, and odometry at about 10 Hz. The "landmarks" are large textured foam blocks. It is not our intent at this time to solve the tracking or landmark recognition problem, so the landmark selection in the omnidirectional images was done by hand, and the visual tracking requires a user to identify targets.

During these tests, the ATRV had a broken wheel. The result was a significant bias in dead reckoning which was partially calibrated out. The error model was also modified to increase the vehicle motion uncertainty to a much higher degree when going through large turns. The invariant filtering algorithm successfully recovers the relationships between the landmarks even with this bias.

Figure 5 and 6 show the time evolution of invariant features d^{01} and d^{23}, the first and fourth features to be added to the relative map. The estimated mean is

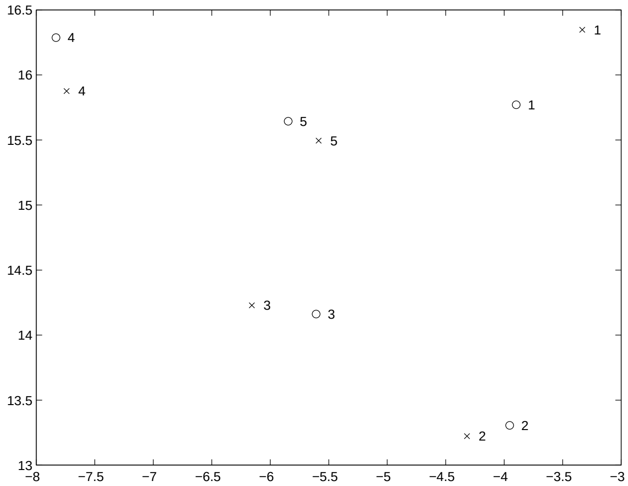

Figure 7: Final estimated map (x's) with MMSE estimate (o's)

shown along with $3\sigma^{ij}$ intervals.

Ground truth for the map is not available. Instead, we implemented a full minimum mean square error (MMSE) batch estimation and used all available data to optimize the estimate of map and robot trajectory. Since the map computed by the invariant filter is not computed with an absolute position reference, there is an unknown planar transformation between the map from the invariant filter and the map from the MMSE estimate. This is accounted for by finding the transformation which most closely aligns the two maps.

The map recovered by the invariant filter is shown in Figure 7 along with the MMSE map. The landmark positions agree to within about a half meter after registration.

6 Summary and Conclusions

In this paper we introduce the invariant filter, a solution to the SLAM problem which decouples the robot position and orientation from the mapping phase in order to simplify computation. The strength of the invariant filtering approach to SLAM lies in the decoupling of the estimates of each environment feature so that the map states are completely uncorrelated with each other and the robot state. This allows much more scalable approach since the computational and memory requirements scale linearly with the number of features. Our implementation has been shown to work onboard a mobile robot in an environment with a modest number of features.

Our current implementation requires some investigation into more reasonable triangulation schemes in order to achieve good maps. The current greedy triangulation does not take geometry into account, and can add relationships which do not adequately constrain the positions of some landmarks.

Acknowledgments

The authors would like to acknowledge the support of NASA through Grant NAG2-1233 and NASA Ames Research Center which has granted a GSRP Fellowship to the first author. We would also like to thank Sanjiv Singh for the use of Bullwinkle the ATRV and Trey Smith for support and help with data collection, and G. Dissanayake for insightful conversations regarding the SLAM problem.

References

[1] Randall Smith, Matthew Self, and Peter Cheeseman. *Estimating Uncertain Spatial Relationships in Robotics*, chapter 3, pages 167–193. Springer-Verlag, 1990.

[2] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th Int. Symposium on Robotics Research*, 1989.

[3] H. Feder, J. Leonard, and C. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research*, 18(7):650–668, July 1999.

[4] John J. Leonard and Hans Jabob S. Feder. Decoupled stochastic mapping. Technical Report 99-1, MIT Marine Robotics Laboratory, Cambridge, MA 02139, USA, 1999.

[5] M. Csorba and H. Durrant-Whyte. A new approach to map building using relative position estimates. *SPIE Vol. 3087*, pages 115–125, 1997.

[6] Paul Michael Newman. *On the Structure and Solution of the Simultaneous Localization and Mapping Problem.* PhD thesis, Australian Centre for Field Robotics, University of Sydney, 1999.

[7] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1988.