

Experimental comparison of techniques for localization and mapping using a bearing-only sensor

Matthew Deans and Martial Hebert
Carnegie Mellon University
Pittsburgh, PA, USA
{mdeans,hebert}@cmu.edu

Abstract:We present a comparison of an extended Kalman filter and an adaptation of bundle adjustment from computer vision for mobile robot localization and mapping using a bearing-only sensor. We show results on synthetic and real examples and discuss some advantages and disadvantages of the techniques. The comparison leads to a novel combination of the two techniques which results in computational complexity near Kalman filters and performance near bundle adjustment on the examples shown.

1. Introduction

In this paper we will present a comparison and experimental evaluation of techniques for localization and mapping for a mobile robot equipped with a bearing-only sensor. The robot has no *a priori* knowledge of the environment (no map) and can only observe its egomotion through odometry and the bearings to landmarks in the environment with an omnidirectional camera. The robot has no external position or heading reference, and cannot measure the range to landmarks.

2. Related work

Much of the work published in Simultaneous Localization and Mapping (SLAM), also referred to as Concurrent Localization and Mapping (CLM), makes use of active range sensors. Approaches generally rely on recursive filtering using a Kalman filter or some variant [1]. There have also been attempts to find optimal estimates using batch techniques [2] or to deal with large environments using a hierarchical approach [3, 4].

The Computer vision literature contains a large body of work on multi-frame Structure From Motion (SFM). Bundle adjustment is a batch technique that is widely used for SFM under perspective projection [5, 6] and is applicable to SLAM. Some attempts have also been made to recursively compute structure and motion using Kalman filters [7, 8].

The SLAM and SFM problems contain features relevant to localization and mapping with a bearing-only sensor. The goal of this paper is to present some comparisons on how techniques from SLAM and SFM perform on the bearing-only SLAM problem including results from experiments on simulated

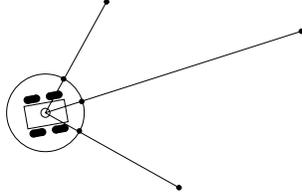


Figure 1. Bearing-only sensor projects landmark position onto unit circle

and real data. To date, little has been published in terms of direct experimental comparison of existing techniques for solving this problem.

3. Localization and Mapping

For simplicity we will only consider a 2D planar environment. The robot position at time i is $\mathbf{m}_i = (x_i, y_i, \theta_i)^T$ and the feature locations are $\mathbf{x}_j = (x_j, y_j)^T$. The odometry is modelled as a stochastic measurement of motion between two consecutive robot poses such that

$$\mathbf{d}_i = f(\mathbf{m}_{i-1}, \mathbf{m}_i) + \nu_i \quad (1)$$

where ν is $\mathcal{N}(0, R_d)$ noise. The bearings are modelled as stochastic measurements of the bearing towards each visible landmark in the rover coordinates,

$$z_{ij} = h(\mathbf{m}_i, \mathbf{x}_j) + \omega_{ij} \quad (2)$$

where ω is $\mathcal{N}(0, \sigma_\omega)$ noise.

3.1. SLAM as 2D Structure from Motion

The Structure from Motion (SFM) work in computer vision provides insight into what we can expect from bearing-only SLAM. In SFM the 3D structure of an environment is reconstructed using 2D projections (images) from unknown camera locations. The perspective camera projection model is very similar to a bearing-only sensor; depth information is lost in the projection. The only knowledge of the scene from a single image is that scene points lie somewhere along a ray originating at the camera center of projection and passing through the image plane at the point where the feature appears. In bearing-only SLAM, landmarks are projected onto a unit circle around the robot (Figure 1), and a similar constraint applies for the ray intersecting this projection surface.

If only the bearing measurements are available, and not the odometry, then a simple counting argument gives us a necessary condition for when the 2D structure from motion problem can be solved. For each robot position there are 3 unknowns, and for each landmark 2 unknowns. If there are M robot positions and N landmarks, then there are $3M + 2N$ unknowns. Assuming that all landmark features are observable from all robot positions, there will be MN measurements, and in general these will be linearly independent. There are 4 gauge freedoms[9] in the estimation, i.e. we can translate, rotate, and scale the solution and the measurements will be unchanged. We must set an

absolute coordinate frame and absolute scale with 4 additional constraints. We can find a solution to the problem when $MN + 4 > 3M + 2N$, or

$$(M - 2)(N - 3) > 2 \tag{3}$$

This constraint on the solution to the problem shows that there is no solution to the two frame SFM problem in 2D, regardless of methodology. Only when there are three robot positions and at least five landmarks or more than three robot positions and at least four landmarks can we get a solution, and this solution is only defined up to the gauge freedoms. In contrast, aligning just two scans from an accurate range-bearing sensor can give an estimate of vehicle motion and make up for errors in odometry.

In most SFM algorithms, little or no a priori information about the camera motion is used for reconstruction, camera motion is recovered in the estimation along with scene structure using only the image measurements. Odometry can be seen as providing a prior for the camera motion which is absent in general SFM problems. If we add odometry then we can immediately disambiguate the scale. There is even a unique solution to the two frame problem if odometry is available since it will provide a direct estimate of the relative pose of the second view. However until the third view is incorporated, bearing measurements cannot begin to correct odometry errors within the estimation algorithm. This can cause initialization problems with Kalman filters which are observed in practice, particularly when odometry is poor.

A final note is that bias in odometry cannot always be corrected in bearings-only SLAM. If the measurements of distance travelled are biased by a scale factor then the estimate of robot motion *and* landmark positions will be scaled by the same factor and no estimator can recover the bias using the bearing measurements. This is an inherent ambiguity. In contrast, SLAM with range-bearing sensors does not suffer from this problem since the range measurements provide direct information about the scale of the solution.

4. Estimating parameters

Finding an estimate for the model parameters $\mathbf{m} = (\mathbf{m}_0^T, \mathbf{m}_1^T \dots \mathbf{m}_M^T)^T$ and $\mathbf{x} = (\mathbf{x}_0^T, \mathbf{x}_1^T \dots \mathbf{x}_N^T)^T$ can be done in many ways. In this paper we wish to make some comparisons between two of these methods in particular. The first method is the Extended Kalman filter and the second is bundle adjustment. Each of these methods has properties which make it attractive for use in SLAM, and the nature of the SLAM problem is important to the way in which these techniques are used and the results that can be expected.

4.1. Extended Kalman Filter

The Extended Kalman Filter (EKF) has been used in SLAM and SFM to recursively estimate a state vector which consists of the current rover pose or camera parameters and the positions of all landmark or scene features [1, 8]. As a new measurement is made, the filter goes through prediction and update steps which incorporate the new measurement and generates a best estimate (mean) and an uncertainty estimate (covariance) for the current rover pose

and the map. The mean and covariance are kept as a sufficient statistic for the posterior probability over state space, and old observations are discarded after being incorporated. Nonlinearities in bearing-only SLAM cause difficulties to the EKF; in particular, getting started is tricky. The posterior over robot pose and landmark position is significantly non-Gaussian such that the mean and covariance in the filter are not a reasonable sufficient statistic for the data. For this reason, we use bundle adjustment (to be explained in the next section) to get an initial state estimate and initial covariance matrix using some portion of the data, then recursively process the rest of the data with a Kalman filter. The quality of the result depends on how much of the data is used to get the initial estimates.

Space considerations prohibit detailed explanation of the use of Kalman filters in SLAM but there exists a large body of literature on the topic, particularly for problems where range and bearing measurements are available [1, 10]. Although they should be included in a complete treatment on this topic, we will also omit discussion of modifications to the Kalman filtering method such as implicit Jacobian computations [11] and covariance intersection [12].

4.2. Bundle Adjustment

Bundle adjustment is a full nonlinear optimization which does not rely on a mean and covariance as a sufficient statistic for previous observations and state estimates. Instead it linearizes the estimation problem at every step using all available observations and the current best estimate. Bundle adjustment may optimize all motion and structure parameters at every step, so the state vector is the entire history of robot pose (trajectory) and the entire map.

We wish to minimize the cost function

$$E(\Theta) = \frac{1}{2}(\mathbf{d} - \mathbf{f}(\Theta))^T \mathbf{R}_d^{-1} (\mathbf{d} - \mathbf{f}(\Theta)) + \frac{1}{2}(\mathbf{z} - \mathbf{h}(\Theta))^T \mathbf{R}_z^{-1} (\mathbf{z} - \mathbf{h}(\Theta)) \quad (4)$$

where $\Theta = (\mathbf{m}^T, \mathbf{x}^T)^T$ is the vector of parameters to be estimated, \mathbf{d} and \mathbf{z} are all odometry and all bearing measurements stacked into vectors, and $\mathbf{f}()$ and $\mathbf{h}()$ are all predicted odometry and bearing measurements stacked into vectors. The first term in (4) penalizes robot motion that does not agree well with odometry measurements and the second term penalizes robot motion and landmark map combinations that do not agree well with bearing measurements.

Taking the first and second derivatives of (4) we find

$$\nabla_{\Theta} E = -J_f^T R_d^{-1} \epsilon_d - J_h^T R_z^{-1} \epsilon_z \quad (5)$$

$$\nabla_{\Theta}^2 E = -\nabla_{\Theta}^2 f R_d^{-1} \epsilon_d + J_f^T R_d^{-1} J_f - \nabla_{\Theta}^2 h R_z^{-1} \epsilon_z + J_h^T R_z^{-1} J_h \quad (6)$$

where $J_f = \nabla_{\Theta} f$ and $J_h = \nabla_{\Theta} h$ are the Jacobian matrices of the measurement equations for odometry and bearings, $\epsilon_d = (d_1 - f(x_0, x_1), d_2 - f(x_1, x_2) \dots)^T$ is the difference between measured odometry and odometry predicted by the model parameters, and ϵ_z is the difference between measured and predicted bearings. These last two quantities are analogous to innovations in a Kalman filtering context.

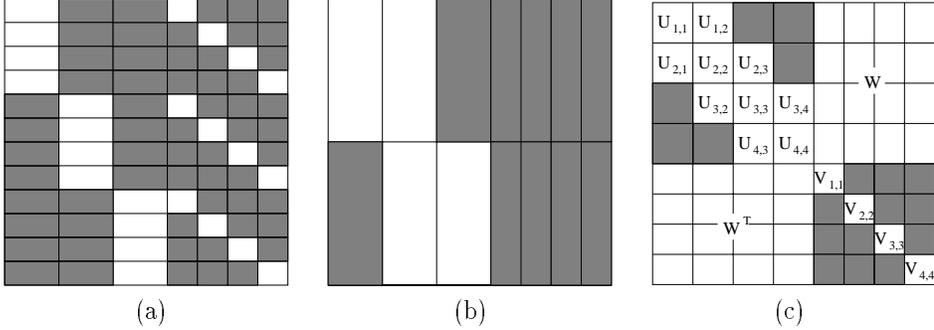


Figure 2. Sparse structure of derivatives. (a) Jacobian of bearing measurement equation. (b) Jacobian of odometry measurement equation. (c) Approximation to the Hessian of the cost function.

When the innovations are small the Jacobian inner products dominate expression (6). The Jacobian term will also dominate when the measurement equations are approximately linear (or linear), since $\nabla_{\Theta}^2 f$ or $\nabla_{\Theta}^2 h$ will be small (or zero). In our case, \mathbf{f} is in fact linear, but \mathbf{h} contains an arctangent. Bundle adjustment drops the second derivative terms and approximates the Hessian of the cost function using only the Jacobian inner products, i.e.

$$H = \nabla_{\Theta}^2 \approx J_f^T R_d^{-1} J_f + J_h^T R_z^{-1} J_h. \quad (7)$$

Note that the true H is the Fisher Information Matrix when Θ is the true parameter vector. We will use it as the inverse covariance matrix for estimating uncertainties. Newton iterations are used to minimize (4) by solving

$$H \Delta \Theta_k = -(J_f^T R_d^{-1} \epsilon_d + J_h^T R_z^{-1} \epsilon_z) \quad (8)$$

and then computing the update

$$\Theta_{k+1} = \Theta_k + \Delta \Theta_k \quad (9)$$

In general solving equation (8) is cubic in the number of model parameters ($3M + 2N$ here). However, photogrammetrists have known for decades how to speed up bundle adjustment using the sparse nature of the structure from motion problem. We can take a similar approach here by exploiting the nature of the bearing-only SLAM problem.

Each bearing measurement depends only on the landmark being measured and the robot pose at the time that the measurement is taken. Therefore, each row of the Jacobian of the bearing measurement equation, J_h , has nonzero entries only for the columns corresponding to the parameters which represent that landmark position and robot pose. The sparse structure is shown in Figure 2(a). The Jacobian of the odometry measurement equation, J_f , depends only on two consecutive robot poses. Each row of the Jacobian of the odometry measurement equation, then, has nonzero entries only for the robot poses before and after the motion. Odometry contains no information about landmark positions and corresponding columns of J_f are zero. The Jacobian has

the structure shown in Figure 2(b). Because odometry and bearing measurement errors are assumed to be uncorrelated, R_d and R_z (and their inverses) are block diagonal. The Hessian $H = J_f^T R_d^{-1} J_f + J_h^T R_z^{-1} J_h$ has the sparse structure shown in Figure 2(c). The upper left is a block tridiagonal matrix U , the lower right is a block diagonal, and the upper right and lower left are rectangular matrices W and W^T . We can write the equation (8) as

$$\begin{bmatrix} U & W \\ W^T & V \end{bmatrix} \begin{bmatrix} \Delta \mathbf{m} \\ \Delta \mathbf{x} \end{bmatrix} = \begin{bmatrix} \epsilon(\mathbf{m}) \\ \epsilon(\mathbf{x}) \end{bmatrix} \quad (10)$$

where the Hessian H , the parameter update $\Delta \Theta$, and the right hand side have been partitioned. Now we premultiply both sides by

$$\begin{bmatrix} I & 0 \\ -W^T U^{-1} & I \end{bmatrix} \begin{bmatrix} U & W \\ 0 & V - W^T U^{-1} W \end{bmatrix} \begin{bmatrix} \Delta \mathbf{m} \\ \Delta \mathbf{x} \end{bmatrix} = \begin{bmatrix} \epsilon(\mathbf{m}) \\ \epsilon(\mathbf{x}) - W^T U^{-1} \epsilon(\mathbf{m}) \end{bmatrix} \quad (11)$$

and solve equations (11) in two steps. We first solve the bottom equation

$$(V - W^T U^{-1} W) \Delta \mathbf{x} = \epsilon(\mathbf{x}) - W^T U^{-1} \epsilon(\mathbf{m}) \quad (12)$$

to find $\Delta \mathbf{x}$ and then substitute it into the top equation, rearranging to get

$$U \Delta \mathbf{m} = \epsilon(\mathbf{m}) - W \Delta \mathbf{x} \quad (13)$$

and solve for $\Delta \mathbf{m}$. Computation of $(V - W^T U^{-1} W)$ is $O(MN^2)$, and solving (12) is $O(N^3)$. Substituting $\Delta \mathbf{x}$ into (13) and solving only requires $O(MN)$ due to the block tridiagonal structure of U , so the overall complexity of bundle adjustment for bearing-only SLAM is $O(MN^2 + N^3)$, which scales linearly with the size of the trajectory. The complexity of bundle adjustment is larger than that of a Kalman filter, which has a complexity of $O(N^3)$ for N landmarks, but it is not complex as a general inverse Hessian approach which might require $O((N + M)^3)$.

5. Results

Results were generated for synthetic and real data. In the synthetic example, we simulated a robot driving in a large circle of radius 100 meters, with 50 landmarks scattered uniformly in a square area circumscribing the robot path. Odometry was simulated by computing the true robot motion and adding Gaussian noise to the estimates of along-track and cross-track motion to simulate slip as well as the rotation. Figure 3 shows estimates of robot motion and landmark map from a Kalman filter and from a bundle adjustment. In each figure, the robot trajectory is the circular path of dots and the landmarks are scattered inside and outside the path plotted with their corresponding uncertainty ellipses.

Both methods were also used to analyze real data from an RWI ATRV. Unfortunately there is no ground truth for this experiment. The robot drove

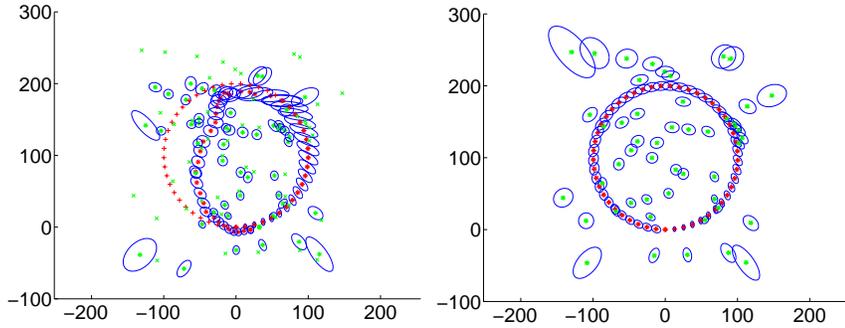


Figure 3. Synthetic data. (a) Kalman filter result (b) Bundle adjustment result

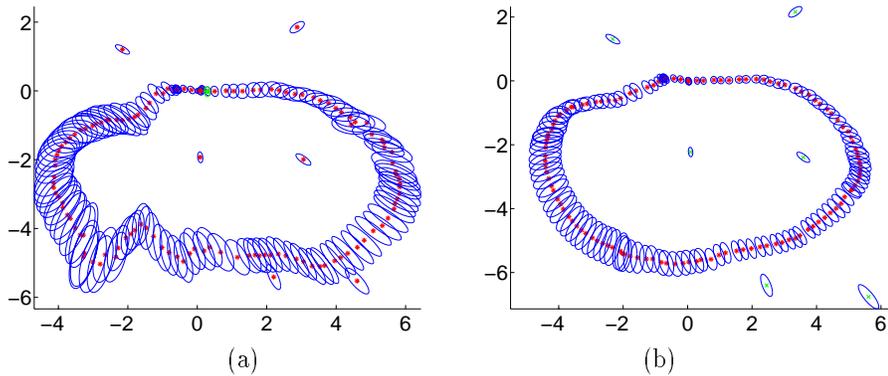


Figure 4. Results on real data (a) Kalman filter (b) Bundle adjustment.

in an approximately elliptical path between six different artificial landmarks. There was a problem with the drive mechanism which caused bias in the odometry measurements. Dead reckoning estimated that the robot drove in a spiral shaped path. Bundle adjustment results are shown in Figure 4(a). Without modelling the bias the bundle adjustment recovers the elliptical path of the robot.

The Kalman filter diverges in the presence of the odometry bias. The odometry was calibrated by fitting a linear function of the data to the bundle adjustment trajectory estimate. With the calibrated odometry and an initial model from bundle adjustment using the first 20 robot poses, the Kalman filter was able to produce the result in Figure 4(b). The lower left portion of the trajectory shows some reconstruction behavior which is most likely due to the introduction of the lower left feature. Since it processes one measurement at a time the Kalman filter does not initialize features well with bearing-only measurements.

5.1. Synthesis

The properties of the methods discussed above and the performance seen in experiments suggests a new approach to bearing-only localization and mapping.

We wish to have a recursive formulation so that the computational burden does not grow with the number of sensor scans made, but we also wish to avoid the bias introduced in the linearization phase in the Kalman filter by considering batches of data rather than individual measurements.

A way to accomplish this is to process data in batches, using observations made during a time interval and computing an estimate for the map and for the pose of the robot during the time interval. At the end of the time interval, the state estimate and all of the data from the time interval is collapsed onto a sufficient statistic, a mean and covariance for the map of landmark features and the last robot pose.

The very first batch of data can be processed in exactly the same way as was described for bundle adjustment. At the end of that estimation, the information matrix will have the same sparse structure as before and can be partitioned as it was in equation (10). We will further partition the U matrix into the block dealing with robot pose from time 1 through $k - 1$, the block dealing with robot pose at time k , and the corresponding off-diagonal blocks as follows

$$P^{-1} = \begin{bmatrix} U & W \\ W^T & V \end{bmatrix} = \begin{bmatrix} U_{1..k-1,1..k-1} & U_{1..k-1,k} & W_{1..k-1} \\ U_{1..k-1,k}^T & U_{kk} & W_k \\ W_{1..k-1}^T & W_k^T & V \end{bmatrix} \quad (14)$$

In order to marginalize this matrix and remove the states $m_1..m_k$, we compute

$$P_k^{-1} = \begin{bmatrix} U_{kk} & W_k \\ W_k^T & V \end{bmatrix} - \begin{bmatrix} U_{1..k-1,k}^T \\ W_{1..k-1}^T \end{bmatrix} U_{1..k-1,1..k-1}^{-1} \begin{bmatrix} U_{1..k-1,k} & W_{1..k-1} \end{bmatrix} \quad (15)$$

The state (m_k, x) and the marginalized information matrix P_k^{-1} are used as a sufficient statistic in the next step of the filter. In general, P_k^{-1} will be a full matrix, it will no longer have the sparse structure of the Hessian from bundle adjustment due to the marginalization step.

As observations are made from new robot positions, the state vector grows to include $\{m_k, m_k + 1, m_k + 2, \dots, x\}$ and the information from the bearing and odometry measurements are added into the information matrix, growing the inverse covariance matrix up and to the left. The upper left block will once again have a block tridiagonal form which can be used to provide fast optimization of the parameters. Since the blockwise elimination and backsubstitution done in equations (10) through (13) fill in the lower right block anyway, the computational complexity is not affected by filling in the block in the previous step. The update step of the new recursive/batch filter is $O(N^3 + kN^2)$ where k is the size of the time window used for the batch update. For small fixed k this is approximately the same computational complexity as the Kalman filter. The algorithm is similar to one reported recently for SFM [13].

Results of applying this filter with a batch size of 10 to the experiment discussed earlier are shown in Figure 5(a). The uncertainty ellipses on each robot pose are plotted as computed at the end of a batch update, so smoothing does not take place over the whole trajectory as it does with full bundle adjustment. Due to this the groups of 10 poses per batch are apparent from the

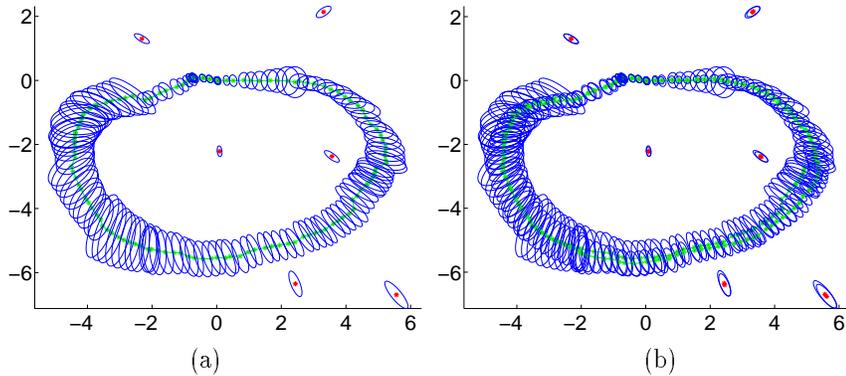


Figure 5. (a) map and trajectory estimated by recursive/batch filter using a batch size of 10. (b) result superimposed with result from bundle adjustment to show agreement.

uncertainty ellipses. Figure 5(b) shows the results from bundle adjustment and recursive/batch methods plotted against each other for comparison. Results for the example considered are very similar for the two methods.

6. Discussion

The examples here illustrate the advantages of bundle adjustment over Kalman filtering for the bearing-only SLAM problem. When Kalman filters are used, they need to be initialized with a good estimate of the robot state and landmark locations and bundle adjustment is an efficient way to compute optimal least squares estimates in order to initialize the filter. Bundle adjustment is empirically more robust to bias in the system model, which is to be expected since the linearizations are recomputed each time and are computed near the optimum. We did not investigate robustness to outliers since there exist both Kalman filters and bundle adjustment techniques specifically designed to deal with outliers[14, 6] but these were not implemented.

For computational and memory requirements, bundle adjustment is not practical for large problems. Because the entire observation history must be recorded, the memory requirement scales as $O(MN)$. By contrast the Kalman filter requires $O(N^2)$ memory. The Kalman filter requires $O(N^3)$ computation for each update. Computation required in bundle adjustment is $O(N^3 + MN^2)$ which grows linearly in the number of robot positions in the estimate. This is significantly less than a general second order least squares optimization but still not practical for a robot operating for long periods of time.

By combining Kalman filtering and bundle adjustment concepts, a new recursive/batch filter was introduced. The filter allows a flexibility in the batch size, trading off computational requirements and performance. If the batch size is set to 1, the extended Kalman filter results, and if the batch size is unlimited, full bundle adjustment results.

References

- [1] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS '91*, pages 1442–1447, 1991.
- [2] F. Lu and E. Milius. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–349, 1997.
- [3] K. Chong and L. Kleeman. Large scale sonarray mapping using multiple connected local maps. In *International Conference on Field and Service Robotics*, pages 278–285, 1997.
- [4] John J. Leonard and Hans Jakob S. Feder. Decoupled stochastic mapping. Technical Report 99-1, MIT Marine Robotics Laboratory, Cambridge, MA 02139, USA, 1999.
- [5] Richard I. Hartley. Euclidean reconstruction from uncalibrated views. In Zisserman Mundy and Forsyth, editors, *Applications of Invariance in Computer Vision*, pages 237–256. Springer Verlag, 1994.
- [6] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In *To appear in Vision Algorithms: Theory & Practice*. Springer-Verlag, 2000.
- [7] T. J. Broida, S. Chandrashekar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Trans. on Aerospace and Electronic Systems*, 26(4):639–656, 1990.
- [8] Ali Azarbayejani and Alex P. Pentland. Recursive estimation of motion, structure and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.
- [9] Philip F. McLauchlan. Gauge invariance in projective 3d reconstruction. In *Proceedings IEEE Workshop on Multi-View Modeling and Analysis of Visual Scenes (MVVIEW'99)*, pages 37–44, 1999.
- [10] M.W.M.G. Dissanayake and et al. An experimental and theoretical investigation into simultaneous localization and map building (slam). In *Proc. 6th International Symposium on Experimental Robotics*, pages 171–180, 1999.
- [11] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the 1995 American Controls Conference*, pages 1628–1632, 1995.
- [12] J. K. Uhlmann, S. J. Julier, and M. Csorba. Nondivergent simultaneous map-building and localization using covariance intersection. In *SPIE Proceedings: Navigation and Control Technologies for Unmanned Systems II*, volume 3087, pages 2–11, 1997.
- [13] P. McLauchlan. A batch/recursive algorithm for 3d scene reconstruction. In *Proceedings of CVPR 2000*, pages II:738–43, 2000.
- [14] R. P. N. Rao. Robust kalman filters for prediction, recognition, and learning. Technical Report 645, Computer Science Department, University of Rochester, 1996.