

# Preliminary Results in Range-Only Localization and Mapping

George Kantor

Sanjiv Singh

The Robotics Institute, Carnegie Mellon University  
Pittsburgh, PA 15217, e-mail {kantor,ssingh}@ri.cmu.edu

## Abstract

*This paper presents methods of localization using co-operating landmarks (beacons) that provide the ability to measure range only. Recent advances in radio frequency technology make it possible to measure range between inexpensive beacons and a transponder. Such a method has tremendous benefit since line of sight is not required between the beacons and the transponder, and because the data association problem can be completely avoided. If the positions of the beacons are known, measurements from multiple beacons can be combined using probability grids to provide an accurate estimate of robot location. This estimate can be improved by using Monte Carlo techniques and Kalman filters to incorporate odometry data. Similar methods can be used to solve the simultaneous localization and mapping problem (SLAM) when beacon locations are uncertain. Experimental results are presented for robot localization. Tracking and SLAM algorithms are demonstrated in simulation.*

## 1 Introduction

In this paper, we present a system of active beacons as a solution to the problem of mobile robot localization. The beacons, which return a range estimate and unique beacon identification number when queried from a mobile robot, form a *de facto* local positioning system when distributed in an environment. The beacons are self-contained, small, and inexpensive. They do not require line-of-sight to the robot, and when used with the methods presented here they do not need to be accurately placed. The result is a low-cost, easily installed system that can be used to localize a mobile robot in both indoor and outdoor environments.

The ability of a robot to localize itself is a fundamental problem for mobile robots. Not surprisingly, many technologies and techniques for robot localization can be found in the literature (eg. [1, 8, 9, 14]). While there are many different variations of the localization problem, we concentrate on three: *static localization*, *position tracking*, and *simultaneous localization and mapping* (SLAM). *Static localization* requires a robot to obtain an accurate estimate of its global position based only on sensor readings. For po-

sition tracking, the robot starts with an initial position estimate that is assumed to be “close” to the actual location. The robot must then keep track of its position as it moves about, using sensory information to continually improve its location estimate and correct for odometry errors [1]. For both position tracking and global localization, it is generally assumed that the robot has a map of its environment, i.e. that the locations of the landmarks used for localization are known. The SLAM problem does not rely on this assumption: the robot must use sensor information to simultaneously localize itself and build a map of its environment. The SLAM problem was first studied by Smith, Self, and Cheeseman [10] in 1990 and has been the subject of significant recent activity [2, 3, 7, 13].

Typically, the work in SLAM supposes that a robot is able to measure both bearing and range to landmarks in the environment [3, 7, 13], but there is some work in trying to use sensors that measure only bearing [2, 12]. This work is related to a body of literature in the computer vision community known as structure from motion, where egomotion and the locations of sparse landmarks are simultaneously extracted from a sequence of images [4]. In contrast to these methods, we examine methods where only range to the landmarks is measured. The most prevalent case of localization using range only is the use of GPS, which has been successfully used for mobile robot localization in outdoor experiments [15]. GPS essentially measures the time taken for signals broadcast from satellites to reach a receiver with the presumption that the locations of the satellites are known with high accuracy. The fact that GPS works only outdoors is a significant drawback. Pseudolites that act as stand-ins for GPS satellites have been used to allow GPS receivers to operate indoors [6], however this solution is undesirable due to the cost and size of the required infrastructure.

For most range-only sensors, the problem of data registration poses a serious obstacle for localization and mapping; the sensors give range to some object without identifying the object. Because beacons in the system we use transmit a unique ID number as well as range, data registration is solved trivially.

In this paper we employ probabilistic methods to generate position estimates from sensor data. We use three methods that have been applied in the past with success: Kalman filtering, Markov methods, and Monte Carlo localization. All three of these methods estimate robot position as a distribution of probabilities over the space of possible robot positions. Originally introduced in 1960, the Kalman filter assumes a multivariate Gaussian distribution [5]. The Kalman filter has the advantage that the representation of the distribution is compact; a Gaussian distribution can be represented by a mean and a covariance matrix. Recent extensions to Kalman filtering allow for non-Gaussian, multimodal probability distributions through multiple hypothesis tracking [8]. The result is a more versatile estimation technique that still preserves many of the computational advantages of the Kalman filter. Markov methods provide another means of estimation [9]. Here, the space of possible robot positions is discretized (often into a “probability grid”) and the probability distribution is approximated by assigning a real number to each point in the discretization. Successive grids or grids arising from independent measurements are combined to create a new grid using Bayes’ rule. Markov methods have the advantage of flexibility, but the size of the discretization can become prohibitive for large areas, small grid resolutions, or problems such as SLAM where the system state has high dimension. Monte Carlo localization provides yet another method of representing multimodal distributions for position estimation [14]. Also known as particle filtering, Monte Carlo localization approximates a distribution using a finite number of weighted samples. The estimated distribution is updated using importance sampling: new samples are drawn from the old distribution at random, propagated in accordance with robot odometry, and then weighted according to available sensor information. One advantage of Monte Carlo localization is that the computational requirements can be scaled as needed by adjusting the number of samples used to represent the distribution.

The paper is arranged as follows: In Section 2, we investigate the problem of robot localization in an environment with known beacon locations using Markovian probability grids. Experimental results are presented. Section 3 extends these ideas to position tracking using Kalman filtering and Monte Carlo localization. Section 4 addresses the problem of localization in an environment with uncertain beacon locations. Here we present a SLAM algorithm that combines intuition with Kalman filtering. Simulation results are given for position tracking and SLAM algorithms.

## 2 Static Localization

In this section, we address the problem of robot localization based solely on current beacon readings. We term this *static localization* because this method does not use

past sensor readings or past estimates of position to determine an estimate of the current position. We assume that the positions of the beacons are known and fixed.

For perfect measurements, determining position from range information is a matter of simple geometry. A robot at distance  $r$  from a beacon must be located on a circle of radius  $r$  centered at the beacon. Determining location from multiple range measurements is just a matter of finding where the corresponding circles intersect. Unfortunately, perfect measurements are difficult to achieve in the real world. The commercially available beacons we use here provide range measurements with an expected error of about 6 feet. We use probabilistic methods to estimate robot position in the face of these uncertainties.

### 2.1 Characterizing Range Measurements

In order to apply probabilistic methods to the localization problem, we first obtain a set of probability distribution functions (pdfs) to characterize the range data provided by the system. Because of measurement discretization and noise, the actual range  $r$  associated with a measurement  $m_i$  is a random variable. We denote the pdf describing the distribution of  $r$  given  $m_i$  as  $p(r|m_i)$ .

The beacon system used in our experiments provides range measurements in feet, discretized to lie in the set

$$\{0, 6, 12, 18, 25, 31, 37, 43, 50\}.$$

A pdf was experimentally determined for each measurement. The resulting pdfs are plotted in Figure 1a.

### 2.2 Creating Probability Grids

The pdfs generated in the previous section give a probabilistic description of the range between the antenna and beacon. In order to be used for robot localization in a plane, these one-dimensional range distributions must be converted to two-dimensional position distributions. Probability grids provide one method of accomplishing this.

To construct a probability grid, the space of interest is discretized into a grid of desired size and resolution. For our purposes, the  $50' \times 50'$  test area was divided into a Cartesian grid of  $1' \times 1'$  squares. Then each square is assigned a real number equal to the probability that the robot resides in that square given a measurement  $m_i$  from a beacon at known location  $x_b$ . We approximate this probability via the following steps:

1. For each square on the grid, compute the value

$$\gamma_s = \frac{p(r_s|m_i)}{2\pi r_s},$$

where  $r_s = \|x_b - x_s\|$  and  $x_s$  is the location of the center of the square.

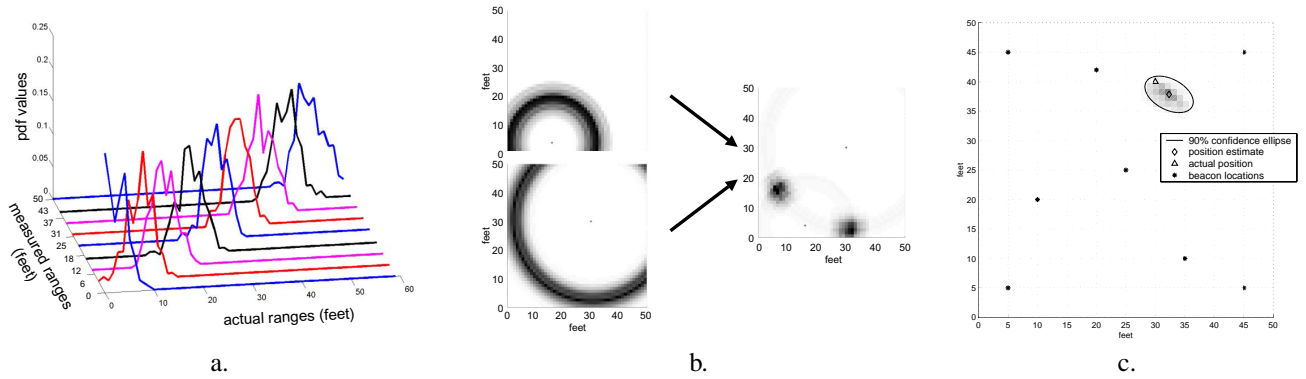


Figure 1: **a.** Experimentally determined pdfs for each of the nine possible range measurements. **b.** Probability grids arising from measurements are shown on the left, the resulting combined probability grid is shown on the right. **c.** Probability grid resulting from the combination of measurements from eight different beacons.

2. Assign to each square the probability

$$P_s = \frac{\gamma_s}{\alpha}$$

where  $\alpha = \sum_{s=1}^{N_s} \gamma_s$ .

Here, step 1 assigns a relative probability to each square while step 2 rescales the relative probabilities to make sure that the overall probability is one.

### 2.3 Combining Probability Grids

Probability grids arising from measurements to multiple beacons can be combined to produce an estimate of robot location. To combine two probability grids, we simply multiply them in a pointwise manner and scale the result so that the sum over the squares is one. Figure 1b depicts this merging process. Note that the number of squares where the robot is likely to be is reduced, providing a better estimate of robot position. This process can be repeated for multiple beacons to yield better results. Figure 1c depicts a probability grid that results from combining measurements from eight beacons.

To get the position estimate  $\hat{x}$  from a probability grid, we take a weighted average of grid locations:

$$\hat{x} = \sum_{s=1}^{N_s} P_s x_s. \quad (1)$$

The covariance matrix  $C$  associated with this estimate is computed as

$$C = \frac{1}{N_s} X X^T, \quad (2)$$

where  $X$  is the  $2 \times N_s$  matrix whose  $s$ th column is

$$X_s = \sqrt{P_s} (x_s - \hat{x}).$$

## 2.4 Experimental Results

We used the technique described above to estimate robot location at approximately 100 points distributed over the  $50 \times 50$  test area with 8 active beacons. The average estimate error over this sample was 1.62 feet. This result is significant considering that the expected error in the range measurements ranges from 5.82 to 7.18 feet.

## 3 Position Tracking

Here we present two methods that take advantage of past position estimates and odometry data to continuously track a robot's position.

### 3.1 Extended Kalman Filtering

When the robot can read data from 3 or more beacons, the resulting combined probability grid usually has only one peak. In these cases, the distribution resulting from the static localization algorithm is reasonably well approximated as Gaussian and we can use a type of extended Kalman filtering algorithm to solve the position tracking problem.

For this discussion, we assume an omnidirectional robot with  $x$  and  $y$  velocities as inputs<sup>1</sup>. We also assume some uncertainty in this model, which conceptually models phenomena like wheel slippage and other unmodeled disturbances. Given robot position  $x(k)$  and inputs  $(u_1(k), u_2(k))$  at time  $t$ , the robot location at time  $t + T$  will be

$$x(k+1) = x(k) + T \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} + \omega(k), \quad (3)$$

where  $\omega(k)$  is an identically independently distributed (iid) Gaussian random vector with zero mean and constant covariance matrix  $R$ .

<sup>1</sup>This assumption is made for clarity. More complicated robot models such as that of a differentially steered robot can be accommodated for with minor adjustments to the approach given here

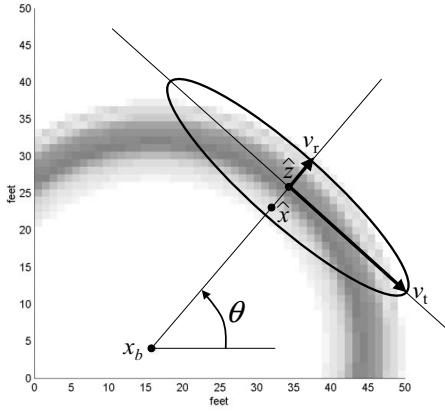


Figure 2: This figure shows how we approximate an annular distribution as Gaussian around an estimate  $\hat{x}$ .

Given a range measurement  $m$  from a beacon located at  $x_b$ , we seek to approximate the annular distribution that results with a Gaussian distribution. Generally, this is not possible, but if we have a prior estimate,  $\hat{x}$  of robot position we can “linearize” the annular distribution around the estimate. The variance in the direction radial to the annulus,  $v_r$ , is chosen to be the same as the variance of the range measurement. The variance in the direction tangent to the annulus,  $v_t$ , is chosen to be very large, reflecting the fact that the range measurement provides little information in that direction. In practice, we choose this variance to be 10 times the variance of the range measurement. The resulting covariance matrix  $C$  to has principal variances  $v_r$  and  $v_t$  in the axial and radial directions respectively. The mean of this approximate distribution,  $\hat{z}$ , is chosen to lie on same radial line with the  $\hat{x}$  with the distance between the beacon and  $\hat{z}$  equal to expected value associated with the measurement  $m$ . This approximation process is depicted graphically in Figure 2. The ellipse plotted in this figure is the variance ellipse associated with  $C$ .

If we let  $\theta$  be the angle between the  $x$  axis and the line through  $\hat{z}$  that points radially away from  $x_b$ , then we can express the mean  $\hat{z}$  and covariance matrix  $C$  of the approximate distribution as

$$\hat{z} = x_b + \begin{bmatrix} \bar{r}_m \cos \theta \\ \bar{r}_m \sin \theta \end{bmatrix}, \quad (4)$$

$$C = \Phi \begin{bmatrix} v_r & 0 \\ 0 & 10 * v_r \end{bmatrix} \Phi^T, \quad (5)$$

where  $\bar{r}_m$  and  $v_r$  are the mean and variance respectively of the pdf associated with the measurement  $m$ , and

$$\Phi = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

In the notation of Kalman filtering, the estimate at the  $k$ th time step is denoted  $\hat{x}(k|k)$  and its associated covariance matrix is  $P(k|k)$ . Given  $\hat{x}(k|k), P(k|k)$ , an input vector  $u(k)$ , and a collection of measurements  $m_i$  from a set of  $N_b$  beacons located at  $x_{bi}, i = 1, 2, 3, \dots, N_b$ . the next estimate  $\hat{x}(k+1|k+1)$ , and covariance  $P(k+1|k+1)$  are computed as follows:

1. Compute predicted next estimate according to robot model:

$$\hat{x}(k+1|k) = \hat{x}(k|k) + T \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}.$$

2. Compute predicted covariance matrix:<sup>2</sup>

$$P(k+1|k) = P(k|k) + R$$

3. Let  $\hat{x}_0 = \hat{x}(k+1|k)$  and  $P_0 = P(k+1|k)$ .

4. For  $i = 1, 2, 3, \dots, N_b$ :

(a) Compute Gaussian approximation  $(z_i, C_i)$  about  $\hat{x}_{i-1}$  using Equation 4 and Equation 5.

(b) Compute new estimate  $\hat{x}_i$  and and covariance  $P_i$  by merging the Gaussian distributions given by predicted estimate and measured estimate [11]:

$$K = C_i (C_i + P_{i-1})^{-1}$$

$$P_i = C_i - K C_i \quad (6)$$

$$\hat{x}_i = z_i + K (\hat{x}_{i-1} - z_i). \quad (7)$$

5. The corrected estimate and covariance are then

$$\hat{x}(k+1|k+1) = x_{N_b}, P(k+1|k+1) = P_{N_b}.$$

To start the algorithm, initial position and covariance estimates ( $\hat{x}(1|1)$  and  $P(1|1)$ , respectively) are found using the static localization method presented in Section 2. In cases where the initialization does not produce a unimodal distribution (eg. when only two beacons are visible), multiple hypothesis testing [8] can be used to solve the tracking problem.

### 3.2 Monte Carlo Methods

In Monte Carlo localization, a probability distribution is represented by a finite collection of samples. The samples, often referred to as *particles*, represent possible robot locations. The basic idea of Monte Carlo localization is to propagate the particles so that they all converge to likely robot locations. Intuitively, the process is not all that different from prediction/update process of Kalman filtering.

<sup>2</sup>For more complicated robot models, this step is more difficult. See [11] for compounding when robot orientation is taken into account

There is a “prediction” step where each particle is propagated according to some robot model, the current inputs, and a random noise selected from an appropriate distribution. There is then an “update” step where the collection of predictions is merged with measurement data. This merging is accomplished through a technique called importance sampling, where particles are weighted according to the pdf associated with the measurement and then resampled. As a result, particles with large weights are likely to be chosen multiple times while particles with small weights are likely not to be chosen at all. In this manner, particles that are in unlikely robot locations are replaced by particles in more likely locations.

Let  $x_p(k)$ ,  $p \in \{1, 2, \dots, N_p\}$ , be a collection of particles at time step  $k$ , where  $N_p$  is the number of particles in the collection. Using the omnidirectional robot model given in Equation 3, the algorithm to propagate this collection based on a measurement  $m(k)$  obtained from a beacon at location  $x_b(k)$  is:

1. Propagate each particle in the collection, i.e. for each  $p \in \{1, 2, \dots, N_p\}$  compute

$$\tilde{x}_p(k) = x_p(k) + T \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} + \omega_p(k),$$

where  $\omega_p(k)$  is generated by a zero-mean Gaussian random number generator with covariance matrix  $R$ .

2. For each  $p$  assign a weight to the  $p$ th particle according to the pdf associated with  $m(k)$  and  $x_b(k)$ :

$$w(p) = p(r_p | m(k)),$$

where  $r_p$  is the distance between  $x_b(k)$  and  $\tilde{x}_p(k)$ .

3. Rescale the weights so that  $\sum_{p=1}^{N_p} w(p) = 1$ .
4. For each  $p$ , randomly choose  $x_p(k+1)$  from the predicted collection. The probability that the particle  $\tilde{x}_i(k)$  is selected during any choice is  $w(i)$ .

### 3.3 Results

Both the Kalman and Monte Carlo methods were tested in simulation using the robot model given in Equation 3 and the set of pdfs determined from experimental data in Section 2. Both systems were simulated with identical beacon locations, beacon returns, and robot trajectories. The covariance matrix for the noise vector  $\omega$  was chosen to be

$$R = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.4 \end{bmatrix}.$$

The results are plotted in Figures 3a and 3b.

We conducted multiple simulations for different robot trajectories. After some transient, the average estimation

error generated by the extended Kalman filter method was 0.73 feet. The result for the Monte Carlo method was 0.93 feet. Kalman filtering requires  $O(N_b)$  computations each step while Monte Carlo requires  $O(N_b N_p)$ . In our experience the smallest  $N_p$  which provided suitable results was about 200, so this difference in efficiency is significant.

## 4 Simultaneous Localization and Mapping

The algorithms presented in Sections 2 and 3 require that the positions of the beacons are known exactly. Algorithms that can cope with uncertain beacon positions will make the proposed positioning system easier to install because the beacons will not need to be placed carefully. The problem of simultaneously determining robot position and identifying the locations of the beacons used to navigate is known as simultaneous localization and mapping (SLAM).

The scenario where initial robot and beacon locations are approximately known is a reasonable one. Good (but not perfect) beacon locations can be obtained through crude measurement or even through estimating location on a building blueprint. Here we adapt the extended Kalman filtering algorithm presented in Section 3.1 to apply it to the SLAM problem. At each step, we use the current estimates of robot and beacon positions together with Equations 4 and 5 to translate each range measurement  $m_i$ ,  $i = 1, 2, \dots, N_b$  into an estimate of the relative displacement between the robot,  $x$ , and the  $i$ th beacon,  $x_{bi}$ . The estimated relative displacement  $\hat{z}_i$  can be thought of as a measurement with zero-mean Gaussian noise with covariance  $C_i$ . The remaining SLAM problem can then be solved according to [10]. Specifically, we define the system state to include both robot and beacon locations, write down dynamic equations for the state (which is easy since the beacons do not move), write the outputs as a function of the state (which are just the differences between robot and beacon locations), and use a Kalman filter to provide an estimate of the state.

Figure 3c depicts the performance of this technique. In this simulation, we assume variance of the range measurement noise is  $v = 1$ . The variances of the initial robot and beacon estimates were set to 25, meaning that the initial guesses have an expected error of 5 feet. In this simulation, the average error of the initial estimate for robot and beacons was 5.13 feet. The average error of the estimates at the end of the simulation improved to 0.77 feet.

## 5 Conclusion

We have presented some robot localization algorithms that employ range only data obtained from active beacons in the environment. Range measurements with expected error on the order of 6 feet were used to generate position estimates with expected error on the order of a foot or less. The beacon technology used in this paper is con-

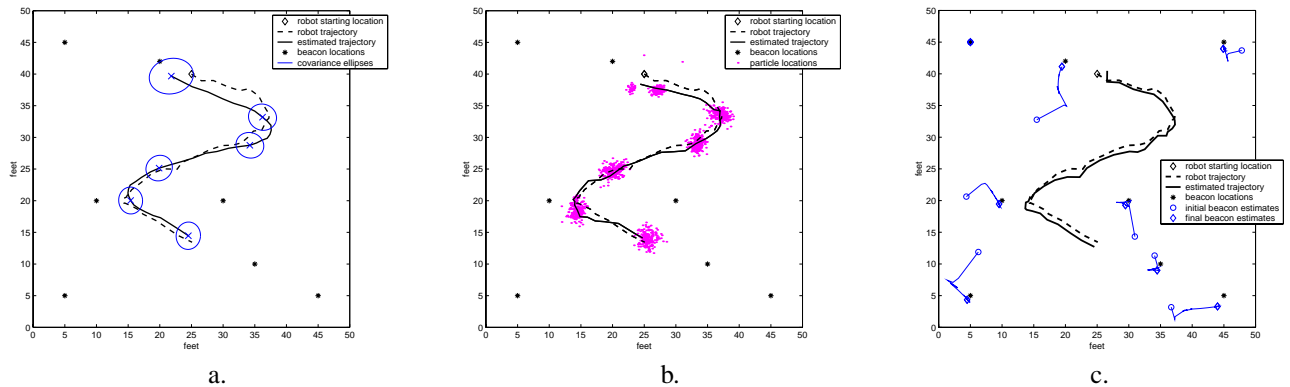


Figure 3: **a.** Results of extended Kalman filter tracking algorithm. **b.** Results of Monte Carlo localization tracking algorithm. **c.** Results of SLAM algorithm for uncertain beacon locations.

tinually improving, one manufacturer has told us that the most recent systems provide an expected error of about one foot. This increased performance should yield future position estimation systems with expected errors on the order of a few inches.

We also presented an algorithm based on Kalman filtering to solve the SLAM problem when the beacon positions are approximately known. Here we use the linearization step described in Figure 2 to put the range-only SLAM problem into a form that can be solved with standard Kalman-based techniques. Conceptually, Markov and Monte Carlo methods should provide a solution for completely unknown beacons, however the high dimension of the SLAM state space renders these methods computationally intractable. Range-only SLAM with completely unknown beacon locations is left as a topic for future research.

## References

- [1] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A.K. Peters, Ltd., Wellesley, MA, 1996.
- [2] M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *Seventh International Symposium on Experimental Robotics*, Honolulu, HI, December 2000.
- [3] M. Dissanyake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba. An experimental and theoretical investigation into simultaneous localization and map building. In *Sixth International Symposium on Experimental Robotics*, pages 265–274, Sydney, March 1999.
- [4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.
- [5] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 8:35–45, 1960.
- [6] E.A. LeMaster and S.M. Rock. Field test results for a self-calibrating pseudolite array. In *Proceedings of Institute of Navigation GPS-2000 Conference*, September 2000.
- [7] J.J. Leonard and H. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Robotics Research: The Ninth International Symposium*, pages 169–176, Snowbird, UT, 2000. Springer Verlag.
- [8] S.I. Roumeliotis and G.A. Bekey. Bayesian estimation and kalman filtering: A unified framework for mobile robot localization. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 2985–2992, April 2000.
- [9] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the IJCAI-95*, pages 1080–1087, August 1995.
- [10] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.
- [11] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, 1986.
- [12] D. Strelow, J. Mishler, S. Singh, and H. Herman. Extended shape from motion to non-central omnidirectional cameras. In *Proceedings of IROS2001*, Maui, HI, December 2001.
- [13] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (Joint Issue)*, 31(5):1–25, 1998.
- [14] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 101:99–141, 2000.
- [15] D. Wettergreen, B. Shamah, P. Tompkins, and W. Whittaker. Robotic planetary exploration by sun-synchronous navigation. In *i-SAIRAS*, June 2001.