SLAM with Expectation Maximization for Moveable Object Tracking

John G. Rogers III, Alexander J. B. Trevor, Carlos Nieto-Granda, and Henrik I. Christensen

Abstract— The goal of simultaneous localization and mapping (SLAM) is to compute the posterior distribution over landmark poses. Typically, this is made possible through the static world assumption – the landmarks remain in the same location throughout the mapping procedure. Some prior work has addressed this assumption by splitting maps into static and dynamic sets, or by recognizing moving landmarks and tracking them. In contrast to previous work, we apply an Expectation Maximization technique to a graph based SLAM approach and allow landmarks to be dynamic. The batch nature of this operation enables us to detect moveable landmarks and factor them out of the map. We demonstrate the performance of this algorithm with a series of experiments with moveable landmarks in a structured environment.

I. INTRODUCTION

Robots need the capability of localization and mapping to perform in many application domains such as mobile manipulation. An essential part of navigation is the ability to remain localized in their environment. In the Simultaneous Localization and Mapping (SLAM) problem, we are interested in estimating the most likely positions of landmarks and the robot given our sensor measurements and control. In the full SLAM formulation, we want to recover the complete robot trajectory, as well as the map (landmark positions) based on our sensor measurements and control. A great deal of research has been done on robotic mapping, and specifically SLAM. An overview of world modeling can be found in [4].

Many SLAM algorithms rely on the assumption that the environment is static, and will perform poorly or fail if mapped objects move. However, to operate in real world dynamic environments, algorithms will need to recognize moveable objects. Consider, for example, a robot that makes a map of a room, and then returns several days later. Some of the features in its map might correspond to immobile objects, such as walls, while some might correspond to objects that may have moved, such as furniture. If someone has moved some of the objects in the room that were part of the robot's map, it will be potentially catastrophic because the SLAM system will make an inconsistent map out of incompatible measurements.

In this paper, we propose an EM approach to data association and static vs. moveable object determination for performing SLAM in a pathological office environment where mapped landmarks move. Our algorithm has been

J. Rogers, A. Trevor, and C. Nieto-Granda are Ph. D. students at Georgia Tech College of Computing {jgrogers, atrevor, carlos.nieto}@gatech.edu

H. Christensen is the Kuka Chair of Robotics at Georgia Tech College of Computing hic@cc.gatech.edu

validated in an indoor environment. This paper is organized as follows: After a discussion of related works in section II, we explain our algorithm for handling moveable landmarks in SLAM by first discussing the Smoothing and Mapping (SAM) algorithm with our extensions in section III. The experimental procedure will be described in section IV and experimental results will be analyzed in sections V and VI. Future directions for this work will be outlined in section VII.

II. RELATED WORK

The summary papers by Durrant-Whyte and Bailey [8], [2] serve as an excellent introduction to the SLAM problem. There are two basic strategies for solving the SLAM problem. The first approach used by the robotics community was based upon the use of the Extended Kalman Filter (EKF) for mobile robot localization. The EKF was used by [5] and [6] for mobile robot localization using an *a priori* map. The first viable SLAM algorithm based on the EKF was reported by [15]. Their technique for addressing the SLAM problem was to augment the state vector with landmark locations.

An alternative technique for solving the SLAM problem is to apply algorithms used in the Computer Vision community for Structure from Motion (SFM). Techniques such as *bundle adjustment* [1] are generating a great deal of interest in the robotics community now due to the availability of high performance computer hardware and the realization that the sparse representations of these techniques can in fact improve performance over the EKF. SFM based techniques typically maintain the full trajectory of the camera and use optimization to find the best trajectory and landmark locations. In robotics, this is known as the *full SLAM* problem since the trajectory is optimized along with the landmark positions.

The full SLAM problem can be represented with sparse matrices by considering the entire robot trajectory within the state vector. Folkesson and Christensen developed Graph-SLAM [9], which uses a nonlinear optimization engine to close loops and avoid linearization errors. Dellaert [7] introduced the Square Root SAM algorithm which uses sparse Cholesky factorization to optimize the robot trajectory and landmarks efficiently. Further progress has been made on online solutions to the SAM problem which uses QR factorization for reordering the measurements to get optimal and online or incremental updates such as with incremental SAM (iSAM) [12], [13].

A common assumption is that the environment being mapped is static. There are two main research directions

which attempt to relax this assumption. One approach partitions the model into two maps; one map holds only the static landmarks and the other holds the dynamic landmarks. Hähnel *et. al.* use an Expectation Maximization (EM) based technique to split the occupancy grid map into static and dynamic maps over multiple iterations in a batch process. This technique is shown in [10] and [11] to be effective in generating useful maps in environments with moving people. Biswas *et. al.* take a finite set of snapshots of the map and employ an EM algorithm to separate the moving components to generate a map of the static environment and a series of separate maps of the dynamic objects at each snapshot. Wolf and Sukhatme [22] [21] [20] are able to separate static and dynamic maps with an online algorithm. Stachniss and Burgard developed an algorithm in [16] which identifies dynamic parts of the environment which engages a finite number of states. The map is represented with a "patch map" that identifies the alternative appearances of the portions of the map which are dynamic, like doors.

The second direction with respect to relaxing the static world assumption is to track moving objects while mapping the static landmarks. Wang *et. al.* in [18] and [19] are able to track moving objects and separate the maps in an online fashion by deferring the classification between static and dynamic objects until several laser scans can be analyzed to make this determination.

Bibby *et.al* [3] use an EM based technique over a finite time-window to perform dynamic vs. static landmark determination; however, their approach differs from ours in several important ways. First, they use a finite sliding window after which no data associations can be changed. Our approach has no such limitation as we are attempting to determine which aspects of the environment are static vs dynamic over the entire mapping run. Bibby's technique does a good job of detecting *moving* objects but it will not be able to detect *moveable* objects over longer time intervals, that might move when they aren't being observed by the robot. Additionally, Bibby addresses the static data association problem by maintaining a distribution of data associations across the sliding window; the data association decision is made permanent at the end of the window (6 steps in Bibby's implementation). An infinite sliding window would be computationally intractable in this implementation due to exponential growth of the interpretation tree; however, our approach offers an alternative solution to the static data association problem which does not suffer from finite history.

III. APPROACH

Our algorithm is based upon the Square Root SAM of Dellaert [7]. We have modified this algorithm with a per-landmark weighting term which enables discrimination between stationary and mobile landmarks to allow for more reliable localization. The remaining landmarks which have a low weight are classified as being moveable and are now tracked by the robot without influencing the robot's trajectory.

A. Square Root SAM

Our implementation of Square Root SAM finds the assignment for the robot trajectory and landmark positions that minimizes the least squares error in the observed measurements. As is common in the SLAM literature, our motion and measurement models assume Gaussian noise. Each adjacent pose in the robot trajectory is modeled by the motion model in equation 1.

$$x_i = f_i(x_{i-1}, u_i) + \nu_i \qquad (1)$$

where $f_i(.)$ is the nonlinear motion model and u_i is the observed odometry from the robot, and ν_i is the process noise. In our case, we use a differential drive robot which has a three-dimensional pose (x_i, y_i, θ_i). The model is shown in equation 2.

$$\begin{pmatrix} \Delta x_i \\ \Delta y_i \\ \Delta \theta_i \end{pmatrix} = \begin{pmatrix} u_0 \cos(\tilde{\theta}) - u_1 \sin(\tilde{\theta}) \\ u_0 \sin(\tilde{\theta}) + u_1 \cos(\tilde{\theta}) \\ u_2 \end{pmatrix} \qquad (2)$$

where u_0 is the forward motion, u_1 is the side motion, u_2 is the angular motion of the robot, and $\tilde{\theta} = \theta_{i-1} + \frac{u_2}{2}$. The measurement model determines the range and bearing to the landmarks. It has the form shown in equation 3.

$$h(x_i, l_j) = \begin{pmatrix} \sqrt{(x_i - l_j^x)^2 + (y_i - l_j^y)^2} \\ \tan^{-1}\left(\frac{(l_j^y - y_i)}{(l_j^x - x_i)}\right) - \theta_i \end{pmatrix} \qquad (3)$$

The linearized least squares problem is formed from the Jacobians of these motion and measurement models as is seen in [7]. By organizing the Jacobians appropriately in matrix A and collecting the innovation of the measurements and odometry in vector b we can iteratively solve for the robot trajectory and landmarks which are stacked in Θ as seen in equation 4.

$$\Theta^* = \arg \min_{\Theta} \| A\Theta - b \|^2 \qquad (4)$$

After each iteration, the Jacobians are re-linearized about the current solution Θ. The solution to this minimization problem can be found quickly by direct QR factorization of the matrix A using Householder reflectors followed by back-substitution. The source paper for this technique [7] exploits sparsity to vastly improve performance; however, we are currently using dense matrices. The optimization currently runs in approximately one second per iteration for a SAM problem of around 50 poses and 100 measurements with dense matrices. We anticipate achieving much better performance once we utilize sparse matrices.

B. Expectation Maximization

To establish an EM algorithm for SAM with moveable objects, we first must express the joint probability model in equation 5.

$$P(X, M, Z) =$$
$$\prod_{poses} P(x_i | x_{i-1}, u_i) * \prod_{landmarks} P(z_k | x_{i_k}, l_{j_k}) \qquad (5)$$

where $P(x_i|x_{i-1}, u_i)$ is the motion model and $P(z_k|x_{i_k}, l_{j_k})$ is the sensor model. We add a hidden variable ω_k to each landmark measurement, which changes the joint probability model to equation 6.

$$P(X, M, Z, \Omega) =$$
$$\prod_{poses} P(x_i|x_{i-1}, u_i) * \prod_{landmarks} P(z_k|x_{i_k}, l_{j_k}, \omega_k) \quad (6)$$

With a Gaussian representation for the sensor model, this new set of parameters ω_k results in the sensor model in equation 7.

$$P(z_k|x_{i_k}, l_{j_k}, \omega_k) \propto$$
$$\exp -(\omega_k((z_k - h(x_{i_k}, l_{j_k})^T \Sigma^{-1}(z_k - h(x_{i_k}, l_{j_k})))) \quad (7)$$

With the interpretation that ω_k is the likelihood that this measurement comes from a static landmark, if the landmark is not static (i.e. $\omega_k = 0$), then the measurement does not affect the joint likelihood since $P(z_k|x_{i_k}, l_{j_k}, \omega_k) = 1$ for all assignments to the robot poses and landmark positions. When the landmark is static (i.e. $\omega_k = 1$), then this weighting term makes this measurement behave like normal. Obviously, the hidden variables ω_k cannot be directly observed by the robot and must instead be estimated from multiple observations of each object. The M step selects new assignments for the ω_k to maximize the joint likelihood. Since the likelihood can be trivially maximized by setting all $\omega_k = 0$, we introduce a Lagrange multiplier to penalize setting too many moveable landmarks. The non-constant portions of the log likelihood for the measurements is now seen in equation 8.

$$l(Z, X, L, \Omega) =$$
$$\sum_{measurements} (-\omega_k(\eta_k^T \Sigma_k^{-1} \eta_k)) - \lambda(1-\omega)^T(1-\omega) \quad (8)$$

where η_k is the innovation of the k-th measurement (the k-th measurement minus its predicted value). This log likelihood is maximized when

$$\frac{\delta l(Z, X, L, \Omega)}{\delta \Omega} = 0 \quad (9)$$

For each ω_k we get the equation 10

$$-(\eta_k^T \Sigma_k^{-1} \eta_k) + 2\lambda - 2\lambda\omega_k = 0 \quad (10)$$

so

$$\omega_k = 1 - \frac{\eta_k^T \Sigma_k^{-1} \eta_k}{2\lambda} \quad (11)$$

We have made the additional modification that the ω_k is not assigned per measurement, but instead since these measurements come from objects we would like to treat the objects as the things that are moveable instead of the measurements being unreliable. This is a simple modification which changes equation 8 into equation 12.

$$l(Z, X, L, \Omega) =$$
$$\sum_{measurements} (-\omega_{l_k}(\eta_k^T \Sigma_k^{-1} \eta_k)) - \lambda(1-\omega)^T(1-\omega) \quad (12)$$

where ω_{l_k} is the weight of landmark l involved in the kth measurement. For each ω_{l_k} we get the equation

$$\sum_{k \in K_l} -(\eta_k^T \Sigma_k^{-1} \eta_k) + 2\lambda - 2\lambda\omega_l = 0 \quad (13)$$

so

$$\omega_l = 1 - \frac{\sum_{k \in K_l} \eta_k^T \Sigma_k^{-1} \eta_k}{2\lambda} \quad (14)$$

where K_l is the set of measurements of landmark l. The Lagrange multiplier λ can be assigned to trade off the penalty for having moveable landmarks. This was the M step of EM.

The E step of EM computes the robot trajectory and the landmark positions with the current estimates of the weighting terms ω_l. The least-squares problem solved by Square Root SAM to find the most likely map is simply the weighted least squares problem, which is to add a weighting term $\omega_l \in [0, 1]$ to each landmark measurement row i.e.

$$H * x_i + J * l_{j_i} = z_{ij} - h(x_i, l_{j_i}) \quad (15)$$

becomes

$$\omega_{l_{j_i}} * (H * x_i + J * l_{j_i}) = \omega_{l_{j_i}} * (z_{ij} - h(x_i, l_{j_i})) \quad (16)$$

where $H = \frac{\delta\nu}{\delta x_i}$ and $J = \frac{\delta\nu}{\delta l_{j_i}}$ with $\omega_{l_{j_i}}$ is the weight assigned to the landmark which we are measuring in this row. In the least squares formulation, this will have the effect of scaling the contribution of this measurement to the overall solution.

C. Moveable Landmark Tracking

After the terminal iteration of the EM algorithm, landmarks which have a weight factor falling below a specific threshold are removed from the SAM optimization and collected in a separate data structure. The final map is optimized once again with the moveable landmarks removed. Measurements on the dynamic landmarks are used with the final trajectory to compute global locations for the dynamic landmarks. These landmarks are now moved into a separate list of moveable landmarks where they could be referred to later to find a list of observed positions. If the moveable landmarks were tagged with semantic information, the robot would then be able to use this data structure as a candidate list of search locations for the object for a retrieval task. The robot can start with the most recently seen position for this object and then try the other places that the object has been seen in the past. Currently, the distribution of positions of the moveable landmarks is being represented as a list of observed locations.

IV. EXPERIMENT

To verify the performance of our algorithm, we performed a series of experiments with moveable landmarks in our office environment.

A. Robot Platform

To test our system, we collected data with a Mobile Robotics Peoplebot. Our robot is equipped with a SICK LMS-291 laser scanner, as well as a Logitech webcam. As measurements, we detect ARToolKit Plus [17] markers in the camera images. The ARToolKit was used in these experiments because the emphasis here is on the detection of static and mobile landmarks. Having landmarks with trivial data association helps us focus on the key contribution of this paper; however, there is no loss of generality and natural landmarks will be considered in future work. The resulting measurements give the relative pose of each marker with respect to the camera. Laser data was also logged, but is only used for visualization purposes. Wheel odometry is also logged, and is used as the input for the motion model.

B. Experimental Setup

The environment used for our experiment was a portion of our lab, consisting of two student offices and the corridors connecting them. ARToolKit markers were placed throughout the environment to serve as landmarks. Some of the markers were pinned to the walls, while others were held by moveable frames which facilitated their movement during the experiments.

C. Procedure

Our first experiments were to move the robot in a circle in one of our student offices measuring 4.5 meters on a side. This office had 12 ARToolKit markers pinned to the walls. In addition to these static landmarks, this office had a total of 10 moveable landmarks which were placed upon the desks and shelves. We performed tests of our implementation of the standard SAM algorithm by moving the robot in this office to collect measurements of landmarks without moving them during the test run. The next experiment was to move the landmarks to a second location within this same cubicle midway through the data collection. We performed a larger scale experiment in which the robot moved between both of our group's student offices. Each of these offices is 4.5 meters on a side, and they are separated by about 12 meters of corridors. We left the 12 ARToolKit markers in the first office from the small scale experiment, and placed 9 markers in the second office. Additionally, we pinned 8 markers to the walls in the corridor between our offices. Several data sets were collected in this setup with varying numbers of moveable landmarks. In each run, the robot was moved in the first office so that each landmark was observed multiple times and then the robot was driven down the corridor. While the robot was being moved down the corridor, the moveable landmarks were transferred from the starting office to random locations in the second office. The robot was then maneuvered in the second office so that each landmark was observed multiple times and then it was driven back to the first office. The robot was driven in the first office in a few loops and was finally placed as close as possible to its starting location. Each test run of this type featured similar trajectories, but always the

Fig. 1. One of the images used as part of our experiments, as taken from the robot.

Fig. 2. The initial state of the map, before optimization. Poses are shown in red, static landmarks are shown in green. The dark green points are laser scans, and are for visualization purposes only.

moveable landmarks were placed in arbitrary positions in the two offices.

The logs were used as input for our algorithm. While the ARToolKit Plus measurements provide the relative pose of the landmark in Cartesian space with respect to the camera, we instead converted this to a range and bearing measurement. As described in section III, the algorithm first considered all measurements, and iteratively adjusted the weights to determine which landmarks were moveable, and which were static. The algorithm iterated until the state converged and the updates were below a specified threshold. Once a stable configuration had been found, the landmarks which had weights below a certain threshold were removed from the SAM problem and were tracked separately. At this point, a final map was generated.

D. Results

We present the longest test run in detail. The initial state of the problem can be seen in Figure 2. This corresponds to
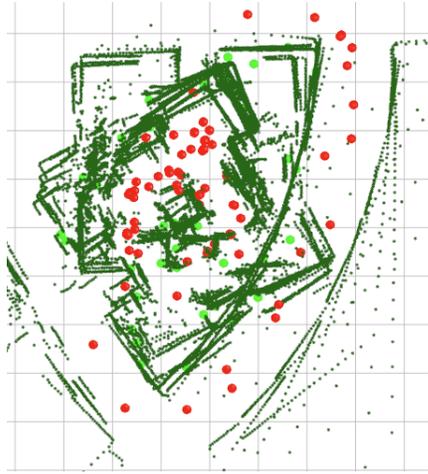
Fig. 3. The resulting map with all measurements (including moveable objects) prior to the first weight assignment.
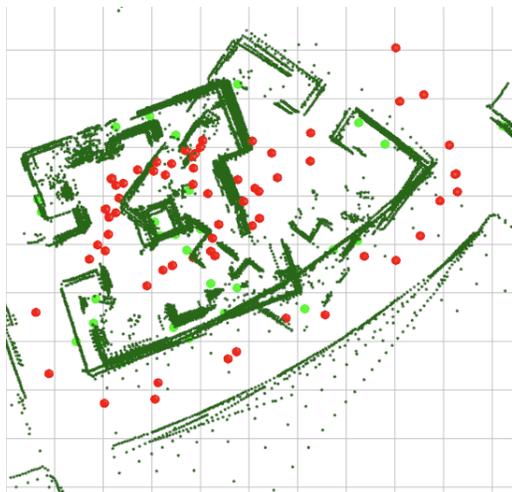

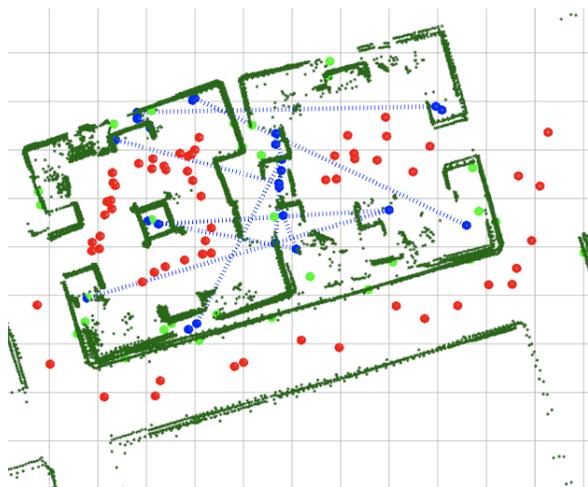Fig. 4. The resulting map after two iterations of the EM algorithm.

Fig. 5. The resulting map after all iterations and thresholding to exclude moveable objects. Poses are shown in red, static landmarks are shown in green, and moveable landmarks are shown in blue, connected by a blue dotted line showing their movement. The dark green points are laser scans, and are for visualization purposes only.

the raw odometry and sensor measurements. The robot starts out in the upper rightmost corner of the left office, facing up in the image. The initial iteration of the EM algorithm will have 1.0 in each ω_k, so each landmark is initially assumed to be static. The SAM optimization is iterated until convergence with these parameters, resulting in a very poor quality map as can be seen in Figure 3. It is apparent in this figure that the moveable landmarks have resulted in incorrect loop closures causing the two offices to intersect almost completely. After two iterations of the EM algorithm the map can be seen in Figure 4. This map is clearly better than the initial state; with two additional iterations of the EM algorithm the low weight landmarks can be thresholded to generate the final map in Figure 5. In this particular run, there were 10 moveable landmarks, 6 of which were detected as moveable. No static landmarks were mistakenly detected as moveable. The remaining 4 moveable landmarks which were mistakenly classified as static were only observed in one of the two offices. Without the observation of the landmark in its second position, the algorithm cannot determine that the landmark had moved. The missing observations can be explained by our use of a webcam and some poor lighting, or the missing landmarks do not appear with a front aspect view which ARToolKit Plus can detect. We have performed two additional test runs of this length and four runs of the single office test, with similar results.

We performed an additional test run where we ignore measurements from the static landmarks. This test was generated by running one of our normal test runs with measurements suppressed from markers that we know to have been static. In this test run, the EM operation was able to correctly identify all of the landmarks as moveable. The final output appears the same as the initial odometry solution. This makes sense because the SAM problem has no measurements between landmarks which affect the trajectory since all of the landmarks had moved.

V. DISCUSSION

While our algorithm worked well for the scenarios we tested, there are some cases that could be more problematic for this technique. Here we provide a brief discussion of some scenarios in which this technique might not perform well.

One case that could cause difficulties for this technique is if several landmarks moved together in a coherent manner. For example, if many landmarks were to move one meter in the same direction, the algorithm might not factor these out, as it might be more likely that the error could be ascribed to poor odometry. This case is of particular interest, because this is what would happen if we were to track many features that were part of a single object. As the object moved, all of the features would undergo the same rigid body transform, moving them in a coherent manner. A possible solution to this would be to track the entire object, instead of individual features on the object.

Because we determine which objects moved based on their residuals, if a landmark were to move only by a small

amount, the residual would probably not be large enough to cause it to be excluded. In this case, it would be used for our algorithm, and would add error to the final map and trajectory.

Another potentially troubling scenario is if most or all of the landmarks we detect are moveable. In our experiment, temporary visual features were used as landmarks for the robot. In practice, some more permanent architectural landmarks should be chosen in addition to potentially moveable landmarks, such as walls.

Also, data association was not an issue because ARToolKit markers were used, and so different markers that appeared near each other were never considered as the same landmark. If we were to use natural features as landmarks, the moveable landmark detection problem becomes much more complex. However, if most of the landmarks we see are static and only a few have moved, a similar technique should still be applicable.

VI. CONCLUSIONS

We have presented an algorithm which relaxes the static world assumption in SLAM. We provided experimental results based upon real robot data and measurements of artificial landmarks. These results serve as a first step towards moveable landmark detection and tracking with real feature measurements and data association. This algorithm should help mitigate the effect of data association errors without the limitation of finite windows for reversible data association.

VII. FUTURE WORKS

The next step in our research will be to remove the reliance on artificial ARToolKit markers. We will support multiple visual and laser feature types to achieve reliable performance with natural landmarks. The use of natural landmarks makes the data association problem a serious issue for our performance; however, we anticipate that our EM algorithm will help mitigate the inconsistency caused by data association errors. The use of the Joint Compatibility Branch and Bound (JCBB) data association test of Neira and Tardós [14] will increase the reliability of the moveable vs static differentiation when static landmarks are in view.

We will be replacing our dense matrix optimization routines with sparse matrices. We will also be investigating incremental SAM [12], [13] to enable online operation. Our EM algorithm currently is run in a loop around the SAM optimization iterations, but we expect that the weights can be recomputed during the SAM iterations. The online operation of this sort of algorithm may also require the use of JCBB [14] as a first step to get a good guess for the weighting term in an initial iteration, permitting faster convergence.

VIII. ACKNOWLEDGMENTS

REFERENCES

[1] R. I. Hartley B. Triggs, P.F. McLauchlan and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. *Lecture Notes in Computer Science*, pages 298–372, 1999.

[2] T. Bailey and H. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part II state of the art. *Robotics and Automation Magazine*, September 2006.

[3] C. Bibby and I. Reid. Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In *Robotics: Science and Systems*, 2007.

[4] W. Burgard and M. Hebert. *Springer Handbook in Robotics*, chapter 36, World Modeling. Springer, 2008.

[5] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. *International Conference on Robotics and Automation*, 1985.

[6] J. Crowley. World modeling and position estimation for a mobile robot using ultra-sonic ranging. *International Conference on Robotics and Automation*, 1989.

[7] F. Dellaert. Square root SAM: Simultaneous localization and mapping via square root information smoothing. In *Robotics: Science and Systems*, 2005.

[8] H. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, June 2006.

[9] J. Folkesson and H. Christensen. Graphical SLAM - a self-correcting map. *International Conference on Robotics and Automation*, 2004.

[10] D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. *International Conference on Intelligent Robots and Systems*, pages 496–501, 2002.

[11] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. *International Conference on Robotics and Automation*, pages 1557–1563, 2003.

[12] M. Kaess, A. Ranganathan, and F. Dellaert. Fast incremental square root information smoothing. In *International Joint Conference on Artificial Intelligence*, 2007.

[13] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 2008.

[14] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, Dec 2001.

[15] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, Winter 1987.

[16] C. Stachniss and W. Burgard. Mobile robot mapping and localization in non-static environments. *Proceedings of the National Conference on Artificial Intelligence*, 20(3), 2005.

[17] D. Wagner and D. Schmalstieg. ARToolKitPlus for pose tracking on mobile devices. *Proceedings of the 12th Computer Vision Winter Workshop*, 2007.

[18] C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. *International Conference on Robotics and Automation*, 2002.

[19] C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. *International Conference on Robotics and Automation*, 2003.

[20] D. Wolf and G. S. Sukhatme. Towards mapping dynamic environments. *International Conference on Advanced Robotics*, 2003.

[21] D. Wolf and G. S. Sukhatme. Online simultaneous localization and mapping in dynamic environments. *International Conference on Robotics and Automation*, 2004.

[22] D. Wolf and G. S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 2005.