2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Applying Domain Knowledge to SLAM using Virtual Measurements

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

Abstract— **Simultaneous Localization and Mapping (SLAM) aims to estimate the maximum likelihood map and robot pose based on a robot's control and sensor measurements. In structured environments, such as human environments, we might have additional domain knowledge that could be applied to produce higher quality mapping results. We present a method for using virtual measurements, which are measurements between two features in our map. To demonstrate this, we present a system that uses such virtual measurements to relate visually detected points to walls detected with a laser scanner.**

I. INTRODUCTION

Localization and mapping are essential capabilities for robots in many domains. For example, many mobile manipulation tasks require a robot to understand its location with respect to its surroundings. The Simultaneous Localization and Mapping (SLAM) problem has been widely studied in order to give robots these capabilities. The SLAM problem can be summarized as the estimation of the most likely robot pose and map, where the map typically consists of landmark poses or an occupancy grid. The full SLAM problem differs in that the goal is to recover the full robot trajectory, as well as the maximum likelihood map based on the sensor measurements and control.

Features detected by the robot can often be related to each other. For example, if we detect a wall using a laser scanner, and detect some visual features at a depth quite near to this wall, it is likely that these features are actually attached to the wall. Another example of a constraint between features is two walls that intersect at a corner and are perpendicular to each other. Introducing these types of constraints into our SLAM problem is a way of injecting knowledge about our environment into our map estimation, and can improve the mapping result. This type of relationship between landmarks can be thought of as a *virtual measurement* between the landmarks.

In this paper, we introduce a method for including such virtual measurements between landmarks, as a way to use our domain knowledge about the environment to improve mapping results. In Section II, we present some related work. Section III discusses our approach to the SLAM problem, including the use of virtual measurements. We then present some results, both from simulation and a robotic platform in Section IV. Finally, a discussion, conclusions, and future work are given in Sections V, VI, and VII.

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

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

This research has in part been sponsored by ARL MAST CTA, the GT-Boeing Manufacturing Initiative, and the KORUS Cognitive Home Robot Project.

II. RELATED WORK

Durrant-Whyte and Bailey [7], [1] provide a thorough introduction to the SLAM problem in general, including many references to popular approaches. One of the most common approaches to the robot localization problem with a given map is the Extended Kalman Filter (EKF), as used in [3] and [5]. Localizing and Mapping concurrently was first successfully demonstrated by [15], where the EKF's state vector was augmented with landmark locations in addition to the robot's pose.

The Computer Vision community has investigated a very similar problem: the Structure from Motion (SFM) problem. In this problem, the goal is to reconstruct the trajectory of a camera as well as the structure of the scene it has observed. This is known as the *full SLAM* problem by the robotics community, in that it estimates the complete robot trajectory instead of just the robot pose.

The robotics community has also approached the full SLAM problem by using sparse matrices, and using a state vector that holds the entire robot trajectory in addition to the landmark poses. A graph based approach to full SLAM was developed by Folkesson and Christensen, and is presented in [8]. Dellaert introduced Square Root SAM in [6], which exploits sparse Cholesky factorization in the optimization problem. SAM has been extended to allow incremental updates in iSAM, presented in [13], [14].

In recent years, we have seen important developments on visual simultaneous localization and mapping (SLAM). Most of the algorithms and techniques use point features in order to reconstruct the environment. In a structured environment, most of the landmarks are planar, quadrangular surfaces, which must be distinguished from the background, typically a poster on a wall, a door-plate or a window. Several contributions propose different solutions to analyze these features and represent them as useful landmarks. Hayet et al. [12] [11] integrated an environment modeling method based on a Generalized Voronoi Graph, relying on laser data and a localization method based on monocular landmark learning and recognition method. Folkesson and Christensen [10] proposed a new feature representation for SLAM called M-Space. The representation addresses feature symmetries and constraints explicitly to make the basic model numerically robust, explicitly modeling the subspace of a feature that has been observed.

There has also been previous work on the use of landmark constraints in SLAM. Choi [4] proposed a geometrically

constrained Extended Kalman Filter (EKF) framework which can estimate line feature positions more accurately as well as allow their covariance matrices to converge more rapidly when compared to the case of an unconstrained EKF handle sparse and noisy sensor data with limited range for an indoor environment. Beevers [2] developed a Rao-Blackwellized constraint filter that infers applicable constraints and efficiently enforces them in a particle filtering framework. Their implementation incorporates prior knowledge of relative rectilinearity constraints between landmarks. While these works discuss constraints between landmarks sensed with range sensors, we address constraints between visual features and range measurements, allowing us to benefit from the advantages of both of these sensor modalities. Additionally, our work uses a graph-based approach in contrast to an EKF or RBPF.

Geometric constraints between features have also been considered by the vision community, for example in the structure from motion problem. Szeliski and Torr considered the relationship between points and planes in [16].

III. Approach

Our technique is based upon the Square Root SAM algorithm of Dellaert [6], but our approach uses the M-space feature representation proposed in [10]. We have also added the capability to add virtual measurements between landmarks.

A. Square Root SAM

Square Root SAM minimizes the error in a set of odometry and landmark measurements in a least squares sense. This is accomplished through the realization that the SLAM problem can be represented as a linear algebra problem for which many efficient algorithms exist. This linear algebra problem relates the measurements between adjacent robot poses with a motion model. The measurements between robot poses and landmark locations are related with a measurement model. We assume that our motion and measurement models are corrupted by Gaussian noise w_i. Each adjacent pose in the robot trajectory is modeled by the motion model:

$$x_i = f_i(x_{i-1}, u_i) + w_i$$

where $f_i(.)$ is the nonlinear motion model and u_i is the odometry measured from the robot. We use a differential drive robot for which the state is (x_i, y_i, θ_i). The model is

$$\begin{pmatrix} \Delta x_i \\ \Delta y_i \\ \Delta \theta_i \end{pmatrix} = \begin{pmatrix} u_0 \cos(\theta_{i-1} + \frac{u_2}{2}) - u_1 \sin(\theta_{i-1} + \frac{u_2}{2}) \\ u_0 \sin(\theta_{i-1} + \frac{u_2}{2}) + u_1 \cos(\theta_{i-1} + \frac{u_2}{2}) \\ u_2 \end{pmatrix}$$

where u_0 is the forward motion, u_1 is the side motion and u_2 is the angular motion of the robot. The measurement model relates the position of the landmark l to the robot. It has the form

$$h(x_i, l_j) = \begin{pmatrix} (l_{jx} - x_i) * \cos(\theta_i) - (l_{jy} - y_i) * \sin(\theta_i) \\ (l_{jx} - x_i) * \sin(\theta_i) + (l_{jy} - y_i) * \cos(\theta_i) \end{pmatrix}$$

We linearize these motion and measurement models around the current estimates for the robot pose and the landmark position. These linearized Jacobians are organized in the measurement matrix A opposite the entries in the state vector which represent the features under consideration. The measurement innovations (the measurement minus the predicted value) are placed in the b vector. This is done exactly as in [6]. The state vector is iteratively selected to minimize the residual of these measurement innovations. At each iteration, the Jacobians are re-linearized to avoid the effects of linearization error during large steps.

$$\Theta^* = \arg\min_{\Theta} \|A\Theta - b\|^2$$

Our solution mechanism for this problem consists of direct QR factorization of the matrix A using Householder reflectors followed by backsubstitution. Dellaert uses sparse linear algebra to achieve faster performance in [6]; however, we are currently using dense matrices. Once the switch is made to sparse matrices, we anticipate achieving real-time performance. For now, our optimization converges in a well conditioned initial state in around one second with a problem of 50 poses and 100 measurements.

B. M-Space

In order to utilize multiple types of features in the map for our SAM problem, we use an approach similar to the M-space representation explained in depth by Folkesson in [10]. The M-space feature representation has previously been demonstrated for graph SLAM in [9], while here we use it in a SAM approach. Instead of updating features in Euclidean space, we represent them using their measurement subspace δx_p. For example, for a line where we cannot observe the endpoints, we would measure the parallel distance and normal direction. A change in a feature's M-space coordinate δx_p can be projected onto the feature's coordinates in euclidean space as δx_f using the feature type's projection matrix $\tilde{B}_f$ according to the following relationship:

$$\delta x_f = \tilde{B}_f \delta x_p$$

By doing this, we can treat walls and points in a consistent manner in our SAM algorithm. Each type of feature (here, points and walls) provides its projection matrix $\tilde{B}_f$, as well as the submatrices of the Jacobian that are specific to the feature type. The innovation η for robot pose x_r and measurement update δx_p is given by:

$$\delta\eta = J_{\eta o} J_{os} J_{sr} \delta x_r + J_{\eta o} J_{of} \tilde{B}_f \delta x_p$$

A detailed explanation of the notation and method is beyond the scope of this paper, but it is explained in depth in [10]. We adapted this representation to SAM by using SAM's J as:

$$J = \frac{\delta h_k(x_{i_k}, l_{j_k})}{\delta x_{i_k}} = J_{\eta o} J_{of} \tilde{B}_f$$

and SAM's H as:

$H = \frac{\delta h_k(x_{i_k}, l_{j_k})}{\delta l_{j_k}} = J_{\eta o} J_{os} J_{sr}$

This formulation allows us to use point features detected with a camera along with wall features detected using a laser scanner in the same graph based formulation.

C. Virtual Measurements

In order to represent relationships between landmarks, we allow for the addition of virtual measurements between two landmarks. These measurements might be to suggest that point landmarks close to walls are coplanar with that wall, or two walls that meet one another are perpendicular. Here, we demonstrate how to add these virtual measurements between a point feature and a wall feature that says the point should lie on that wall.

We identify points that should be constrained to walls by using a threshold in the likelihood of the point being on the wall given its current posterior distribution. In the case of visually detected features, we can set this threshold based on the camera's uncertainty. If we would like to add a constraint that a point (x_c, y_c) should lie on a wall x_l with endpoints (x_{o_1}, y_{o_1}) and (x_{o_2}, y_{o_2}), we have the point to line distance metric:

$$\eta = \frac{-\delta y_o x_p + \delta x_o y_p + \delta y_o x_{o^1} - \delta x_o y_{o^1}}{\sqrt{\delta x_o^2 + \delta y_o^2}}$$

where $\delta x_o = x_{o^2} - x_{o^1}$ and $\delta y_o = y_{o^2} - y_{o^1}$ which are the feature coordinates of the wall. The point feature is (x_p, y_p).

The Jacobian of this measurement with respect to the point coordinates is:

$$\frac{\delta \eta}{\delta x_{point}} = \begin{bmatrix} \frac{-\delta y_o}{\sqrt{\delta x_{o^2} + \delta y_{o^2}}} & \frac{\delta x_o}{\sqrt{\delta x_{o^2} + \delta y_{o^2}}} \end{bmatrix}$$

The Jacobian of this measurement with respect to the wall coordinates is:

$$\frac{\delta \eta}{\delta x_{wall}} = \begin{bmatrix} \frac{-\delta y_o(\delta x_o(x_p - x_{o^2}) + \delta y_o(y_p - y_{o^2}))}{(\delta x_o^2 + \delta y_o^2)^{\frac{3}{2}}} \\ \frac{\delta x_o(\delta x_o(x_p - x_{o^2}) + \delta y_o(y_p - y_{o^2}))}{(\delta x_o^2 + \delta y_o^2)^{\frac{3}{2}}} \\ \frac{\delta y_o(\delta x_o(x_p - x_{o^1}) + \delta y_o(y_p - y_{o^1}))}{(\delta x_o^2 + \delta y_o^2)^{\frac{3}{2}}} \\ \frac{\delta x_o(\delta x_o(x_p - x_{o^1}) + \delta y_o(y_p - y_{o^1}))}{(\delta x_o^2 + \delta y_o^2)^{\frac{3}{2}}} \end{bmatrix}^T$$

These Jacobians represent the $\frac{\delta \eta}{\delta x_{wall}}$ in the global reference frame. This is now multiplied with the $\tilde{B}$ matrix for wall measurements to relate the computed δx_f to a change in the measurement space value δx_p. These Jacobians are inserted into the measurement matrix A so that they relate x_p from the wall and point landmark. The residual distance η is placed in the b vector at this measurement.

With this virtual measurement, the optimization routine will try to pull the point onto the wall, and also it will pull the wall towards the point. Note that this method does not require that the point lie exactly on the line, but instead pulls the point measurement closer to the wall measurement. This allows for objects that lie very near the wall but not actually on the wall.

IV. Experiment

To test our system, we performed two experiments: a simulated experiment so that we can investigate our system while having the benefit of ground truth, as well as an experiment with data from our mobile robot.

A. Robot Platform

To test our system, we collected data with a Mobile Robotics Peoplebot, shown in Figure 1. Our robot is equipped with a SICK LMS-291 laser scanner, as well as an off-the-shelf webcam. As measurements, we detect AR ToolKit Plus [17] markers in the camera images, and extracted line features from laser scans using a Hough transform. The measurements of the AR ToolKit Plus markers give the relative pose of each marker with respect to the camera. Wheel odometry is also logged, and is used as the input for the motion model. Because our algorithm is a batch algorithm, data was logged to a file and processed offline.

Fig. 1. The robot platform used for these experiments. The camera attached to the laptop is the one that is used to detect the AR ToolKit Plus markers.

Fig. 2. The experimental setup. The black and white squares on the cabinets are the AR ToolKit markers used as landmarks.

B. Procedure

The environment used for our experiment was a portion of a corridor in our lab, shown in Figure 2. The corridor has a long wall, on which we placed several AR ToolKit markers. The robot's laser scanner was used to detect lines corresponding to walls in the laser scan, and images from the camera were recorded. We recorded data at 15 poses, and used this as input for our SLAM system.

C. Results

While we do not have ground truth available for the experiment performed with our robot, we can qualitatively evaluate the results. The raw data is plotted and shown in Figure 3. We then performed our SAM optimization as described in Section III, both with and without adding constraints between points and walls. The result with no constraints added is shown in Figure 4, and it can be seen that many of the detected landmarks do not fall on the line detected by the laser scanner. We then performed the SAM optimization again, this time adding constraints that any points detected within $0.4m$ of the wall should be on the wall. The result, shown in Figure 5, shows the points being much closer to the line, resulting in a map that appears more accurate than the result obtained without the constraints.

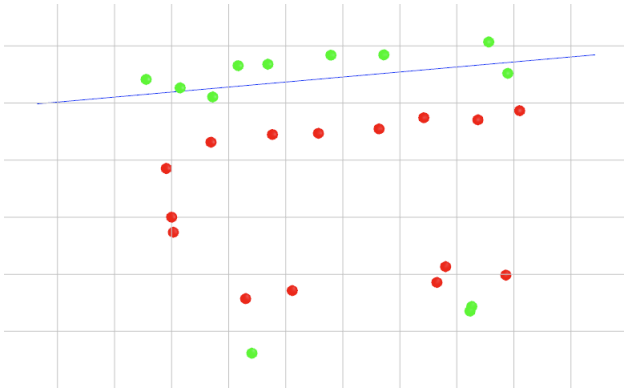

Fig. 3. The initial state of the map, before optimization. Poses are shown in red (dark), static landmarks are shown in green (light), and walls are shown in blue.

D. Simulated Experiment

In order to better evaluate the effects of using this type of constraint in our map estimation, we also performed an experiment in simulation. This allowed us to empirically compare the map estimation both with and without the use of constraints against the simulated ground truth. Robot poses were simulated, with Gaussian noise, according to the motion model described in Section III. Measurements of point landmarks were also simulated, also corrupted by Gaussian noise. Wall landmarks, parameterized by their two endpoints, were also simulated, with the two endpoints being corrupted by Gaussian noise.

The environment we set up was a $6m$ by $7m$ box consisting of four walls, with landmarks placed along the walls, and some within the environment, as shown in Figure 8. The

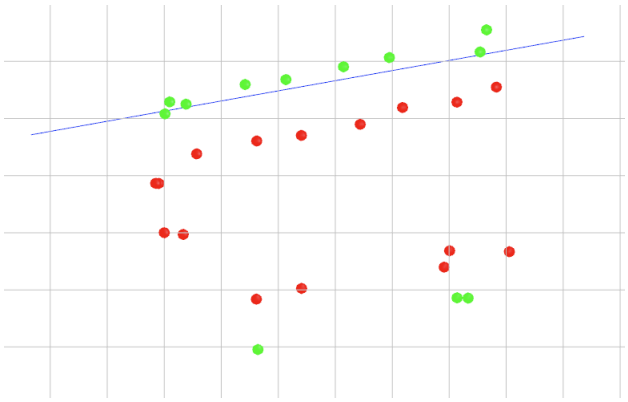

Fig. 4. The map after optimization, but without using constraints between features. Poses are shown in red, static landmarks are shown in green, and walls are shown in blue.

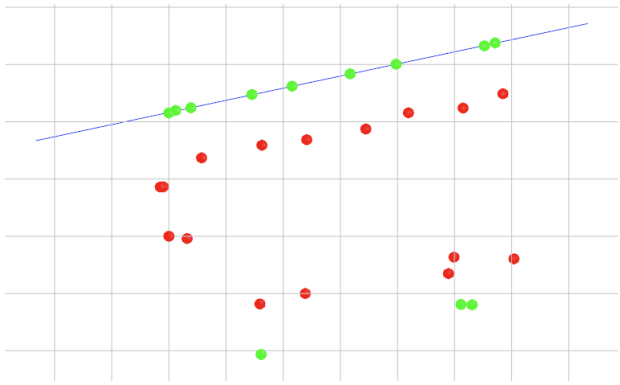

Fig. 5. The map after optimization, including constraints between features. Poses are shown in red, static landmarks are shown in green, and walls are shown in blue.

measurement noise on the line measurements was set to 0.02, while the points had a variance of 0.1. The simulated robot path was a circular trajectory with 40 poses. The experiment was performed both with and without the use of virtual measurements. The raw data is shown in Figure 6, the result without including virtual measurements is shown in Figure 7, and the result with virtual measurements is shown in Figure 8. The average error on the robot poses and landmark poses was then calculated for both cases, as was the variance on the pose error and landmark error. The results are summarized in Table I.

While the pose error was similar between the two cases, the landmark error was reduced when virtual measurements

	Without VMs	With VMs
Average Pose Error:	$0.044654m$	$0.044426m$
Pose Error Variance:	$0.000665m^2$	$0.000727m^2$
Average Landmark Error:	$0.063480m$	$0.043084m$
Landmark Error Variance:	$0.001266m^2$	$0.001044m^2$

TABLE I

SIMULATED RESULTS BOTH WITH AND WITHOUT VIRTUAL MEASUREMENTS, WITH RESPECT TO THE GROUND TRUTH FROM THE SIMULATOR.

were added. We were able to use our domain knowledge that points detected near walls using our noisy point sensor should actually be placed on the wall, which has been measured with a much less noisy wall sensor.

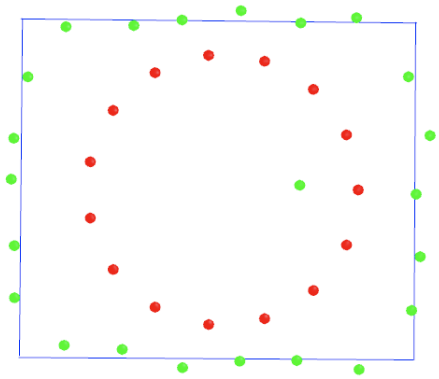

Fig. 6. The initial state of our simulated experiment. Note that the robot's initial belief (shown) is that it moved in a perfect circle. The "actual" poses in the simulator were corrupted by Gaussian noise. Poses are shown in red, static landmarks are shown in green.

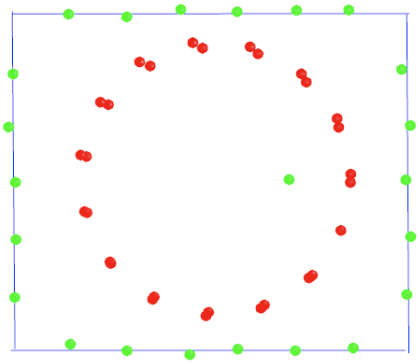

Fig. 7. The simulated experiment after optimization, without inclusion of virtual measurements. Poses are shown in red, static landmarks are shown in green.

V. Discussion

We can consider this approach of adding constraints as a way to use our domain knowledge to improve our map estimation results. Indoor human environments are highly structured, and so they have a variety of constraints that we might wish to encode in our estimation problem. In addition to the collinearity constraint demonstrated here, the general approach used could be applied to add many different types of constraints to our map estimation. As we discussed in the related work, some previous work has focused on adding parallel or perpendicular constraints between walls. This type of constraint could also be added using our approach.

While this approach can improve results when we add virtual measurements that correctly relate two features, one of the shortcomings of this method is that we would introduce significant error into our maps if we were to add virtual measurements between features that should not be related. During informal testing, we tested the effect of adding a virtual measurement between features that were not actually related. As one would expect, this resulted in very high error on the map. However, due to the graph based approach used here, such mistaken virtual measurements can be undone if we re-optimize without them.

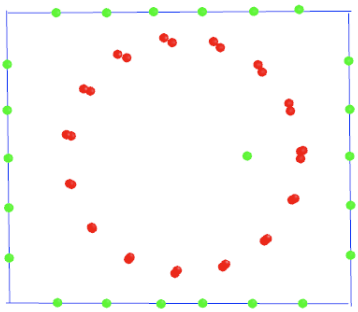

Fig. 8. The simulated experiment after optimization, including constraints between features. Poses are shown in red, static landmarks are shown in green, and walls are shown in blue.

VI. Conclusions

We have proposed an approach to the full SLAM problem that allows for virtual measurements between landmarks to be used in our map estimation. Preliminary experimental results show that this approach might perform well on real robots, and that it can improve the quality of resulting maps when appropriate virtual measurements are added.

VII. Future Works

Here we have considered only two types of features: points and walls. Future efforts certainly include additional types of features, such as visually detected lines, SURF features and object recognition in place of the ARToolkit markers. We demonstrated only one type of virtual measurement here, a constraint between a point and a wall. As mentioned in the discussion, we would also like to consider other types of constraints such as parallelism or perpendicularity between lines by applying a similar approach.

Our approach uses a simple threshold to determine when to apply a virtual measurement, but the problem of when such a constraint is applicable deserves more study. Depending on the type of visual feature detection used, we may be able to reason about whether or not a feature should be constrained to lie on a wall, or not. For example, if we know our visually detected features come from a pattern on the wallpaper, a poster, or a sign, then it would be appropriate to use a virtual measurement. However, if the feature comes from a wall clock or a fire alarm, then such a constraint should not be used.

Additionally, our system thus far has been in 2D. However, we would like to extend our work to 3D, to consider features such as planes or quadrangular objects and to estimate the 6D robot trajectory.

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge Jinhan Lee for his help with the ARToolKitPlus fiducial recognition and the helpful comments of our reviewers. This research has in part been sponsored by ARL MAST CTA, the GT-Boeing Manufacturing Initiative, and the KORUS Cognitive Home Robot Project.

REFERENCES

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

[2] Kristopher R. Beevers and Wesley H. Huang. Inferring and enforcing relative constraints in SLAM. In *2006 Workshop on the Algorithmic Foundations of Robotics (WAFR 2006)*, New York, NY, July 2006.

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

[4] Young-Ho Choi, Tae-Kyeong Lee, and Se-Young Oh. A line feature based slam with low grade range sensors using geometric constraints and active exploration for mobile robot. *Auton. Robots*, 24(1):13–27, 2008.

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

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

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

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

[9] J. Folkesson, P. Jensfelt, and H. I. Christensen. Graphical slam using vision and the measurement subspace. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005)*, pages 325–330, 2005.

[10] John Folkesson, Patric Jensfelt, and Henrik I. Christensen. The m-space feature representation for slam. *IEEE Transactions on Robotics*, 23(5):1024–1035, 2007.

[11] J.B. Hayet, C. Esteves, M. Devy, and F. Lerasle. Qualitative environment modeling with cooperating visual and range sensors. In *Proc. Int. Conf. on Intelligent Robots ansd Systems (IROS'02)*, Lausanne, 2002.

[12] Jean-Bernard Hayet, Frédéric Lerasle, and Michel Devy. A visual landmark framework for mobile robot navigation. *Image Vision Comput.*, 25(8):1341–1351, 2007.

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

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

[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] R. Szeliski and P.H.S. Torr. Geometrically constrained structure from motion: Points on planes. *Lecture notes in computer science*, pages 171–186, 1998.

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