Integrated systems for Mapping and Localization

Patric Jensfelt, Henrik I Christensen & Guido Zunino
Centre for Autonomous Systems, Royal Institute of Technology
SE–100 44 Stockholm, Sweden
{patric,hic,guidoz}@nada.kth.se [1]

1 Introduction

A fundamental component of any mobile robot system is methods for localization and navigation. Almost all deliberative tasks that a robot must carry out has as an underlying assumption that the system can answer the three questions: ”Where am I?”, ”Where am I going?”, and ”How do I get there?” [1]. The first question is termed localization, while the second questions implicitly is termed localization and/or place recognition, while the third question is termed path planning.

The localization problem can roughly be divided into three parts, *pose initialization, pose tracking and map acquisition*. This separation is to a large extent based on the way people have approached the problem. The first two problems are often what people mean when they talk about localization, the latter is referred to as mapping. Studying the problem in more detail soon makes it clear that localization and mapping cannot be separated. Since localization requires a map and mapping requires known pose, there is a “chicken and egg” problem. Which comes first? The answer to the question is that they have to be carried out concurrently. The process of building a map at the same time as estimating the pose of the robot is called *Simultaneous Localization And Mapping* (SLAM) or *Concurrent Mapping and Localization* (CML). SLAM is different from ordinary map acquisition since the uncertainty in the robot pose is accounted for when building the map. The correlation between the estimate of the robot pose and the map that is being constructed is thus explicitly modeled.

Three main directions can be identified in the literature for SLAM: topological, grid-based and feature-based approaches. In topological techniques the environment is modeled as a graph, in the extreme case completely without geometric information. Localization is achieved by recognizing places/nodes. Topological mapping scales well to large environments since the amount of information that is stored is limited to the description of the places/nodes. One of the major disadvantages with topological SLAM is that it typically is quite difficult to reliably recognize a place.

Ever since the introduction of the occupancy grid by Moravec and Elfes [2], the grid based mapping techniques have been widely used for mapping and localization. A typical implementation of grid-based SLAM is to keep a local and a global grid. The global grid is where the overall map is stored and the local map is used to update it. By matching the local map to the global map a measurement is given of the position of the robot. The local map can also be used to improve the global map. An inherent problem with grid based methods is that they are computationally expensive and consume much memory.

The idea of trying to extract features from the environment is quite natural, this is for example how most city maps are constructed. Extreme examples of features are labels on doors, which specify the room it is leading to, but other less discriminative features can also be found. In a structured environment, which most office environments are examples of, lines, corners and edges are common features. The features can be parameterized by, for instance, their color, length, width, position, etc. A feature based map can in general be written

$$\mathcal{M} = \{f_j \mid j = 1,\ldots,N\}, \tag{1}$$

where f_j is a feature and N is the number of features in the map.

Leonard, Durrant-Whyte and Cox are quite firm in their belief that feature-based methods are the way to go, when they say: “*we believe that navigation requires a feature-based approach in which a precise, concise map is used to efficiently generate predictions of what the robot should “see” from a given location*” [3].

Localization using artificial landmarks is well understood and reliable, but requires modified environments. When using natural landmarks for localization, one of the problems is to find suitable candidates for such. In [4] a polygonal map is used to predict the readings from a laser range finder. Drumheller also uses the line primitive for doing sonar based localization [5]. Arras and Tomatis complement horizontal lines extracted from laser range data with vertical lines extracted from vision [6]. Leonard and Durrant-Whyte use natural geometric beacons in [7]. Examples of geometric beacon are lines, corners and edges. The beacons are extracted from densely sampled sonar data.

Historically, papers dealing with mapping have said little about how the map is later used for localization. In many applications the robot is required to make a map of the environment the first time it visits an area. At this point mapping is the primary objective and other tasks will have lower

[1] This research has been sponsored by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems. The funding is gratefully acknowledged.

priority and will thus not be given computational resources. However, once the area has been covered and a map has been constructed the other tasks will be important. To reduce the computational cost the localization part of the system should shift to maintenance mode, both for mapping and pose estimation. To achieve this we argue that it is an advantage to chose a map representation that focuses on capturing the highly static part of the environment. This way the need for map maintenance will not be as great.

Most approaches reported in the literature operate in a continuous mode SLAM. To cope with error situations it is however necessary to consider detection of “track failure” and “initialization”. In principle the operation of a SLAM method can be divided into “global pose estimation” (for initialization and recovery), “error detection” and normal operation. There is consequently a need to embed SLAM methods into a complete framework that allow integration of all three modes into a unified model.

In this paper the basic SLAM problem in introduced (Section 2). Subsequently it is outlined how multiple representations are integrated to provide an efficient framework for mapping (Section 3). In this context the issue of error recovery is discussed (Section 4). With a suitable framework in place the use of multiple sensory modalities is outlined (Section 5), before a number of systems used for evaluation are outlined (Section 6). These systems have been used for extensive experiments, of which some basic results are described in Section 7. Finally the overall concept is discussed and future issues of research are outlined (Section 8).

2 The Basic SLAM Problem

The basic problem of SLAM can be defined as constructing a map, M, of the environment while the robot is moving and at the same time using M to do localization. This problem can be casted as a traditional estimation problem using a Least Square minimization approach. When formulated as a recursive system it is natural to use a Kalman basis as outlined below. A notorious problem is computational complexity which calls for careful consideration of methods for management of complexity.

2.1 The Kalman formulation

Most of the work on feature-based SLAM can be traced back to [8], where *stochastic mapping* is presented, which is an extended Kalman filter based approach to SLAM. The robot pose and the location of all map features are collected in one large state vector. Both the robot pose and the location of the map features are updated when mapped features are re-observed. In essence, localization is performed within the current map, and when the robot enters new areas the state vector is augmented with new features. The two main steps of stochastic mapping are prediction and update. In the prediction step the control signals to the robot or odometric information is used to predict the state at the next time step. In the update step measurements of features are used to update the robot pose and the mapped features.

The state vector, $\mathbf{x}_k$, incorporates the location of all N mapped features, $\{\mathbf{x}_k^i,\ i = 1,\ldots,N\}$ as well as the robot pose and is thus given by

$$\mathbf{x}_k = \begin{pmatrix} \mathbf{x}_k^r & \mathbf{x}_k^1 & \ldots & \mathbf{x}_k^N \end{pmatrix}^T . \qquad (2)$$

The SLAM problem can, in the feature-based setting, be formulated as augmenting and estimating $\mathbf{x}_k$ given measurements of the environment, i.e. estimating both $\mathbf{x}_k^r$ and $\{\mathbf{x}_k^i,\ i = 1,\ldots,N\}$ and adding new features if needed.

Let $\hat{\mathbf{x}}_{k|k}$ denote the estimate of the state vector at time k. The corresponding estimation error covariance matrix can be decomposed as

$$P_{k|k} = \begin{pmatrix} P^{rr} & P^{r1} & \ldots & P^{rN} \\ P^{1r} & P^{11} & \ldots & P^{1N} \\ \vdots & \vdots & \ddots & \vdots \\ P^{Nr} & P^{N1} & \ldots & P^{NN} \end{pmatrix}, \qquad (3)$$

where P^{rr} is the covariance matrix of the robot pose estimate, and P^{ii} the covariance matrices for the features. The correlations between different features and the robot pose are given by the off-diagonal sub-matrices. The estimated state vector $\hat{\mathbf{x}}_{k|k}$ together with the covariance matrix $P_{k|k}$ is the stochastic map. This term highlights the fact the the map is not fixed, it is being estimated as the robot moves along.

The equations covering the stochastic map in its standard form can be found in for example [7].

3 From theory to practice

3.1 Embracing the pros of the topological map

An inherent problem with a brute force implementation of SLAM is the curse of dimensionality. The computational cost grows quadratic with the number of features in the map. This limits a typical implementation to have in the order of hundred features[9]. In [10] a technique called decoupled stochastic mapping in introduced. The idea is to take advantage of the topological representation and let the world be represented by a set of sub-maps. The computational cost can then be adjusted by selecting the size of the sub-maps.

3.2 Exploration Strategy

Burgard et al. [11] addressed the problem of exploration with a grid based approach, achieving good results by basing the decision of where to explore on minimization of the expected future entropy of the hypotheses distribution. Unfortunately the required processing power is proportional to the size of the area and to the resolution of the grid. In [12] it is instead proposed to use a feature-based approach, only considering the

Figure 1: The SuperScout Louie with a PTU mounted SICK.

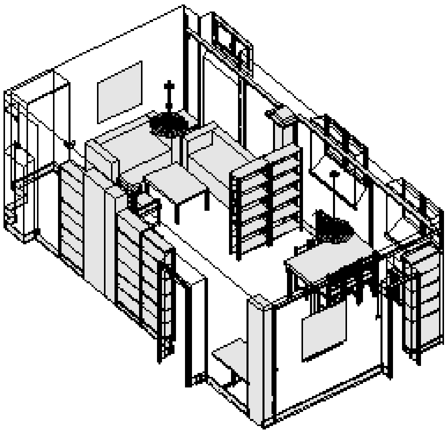

Figure 2: 3D model of the living-room at CAS.

relevant part of the map when taking a decision. This eliminates the problem of being dependent of the map size and is thus less costly in processing power.

3.3 Expanding the sensor view

In a typical indoor environment you will find that many of the stable structures that result in features in the map are present at different heights and not only at the height of the sensor. The lines associated with walls is one such example, door frames is another example. What is occluded at sensor height is often free higher up. By mounting the laser sensor at a pan-tilt unit this can be utilized. One of our Nomad Scout robots equipped accordingly is shown in Figure 1.

4 Techniques for error detection and handling

Anyone who has done real world experiments knows that sometimes things do not always go according to plans. For a truly autonomous system to handle these situations, it must have means to counter act. The problem is two-fold, detecting that an error has occurred and taking the appropriate action, both which are tough problems.

In an EKF approach to localization or mapping there are typically two main courses of failure, one is erroneous data associations between sensor data and the map the other is large unexpected perturbations of the robot. When dealing with mapping in the EKF frame work, map slippage must also be considered.

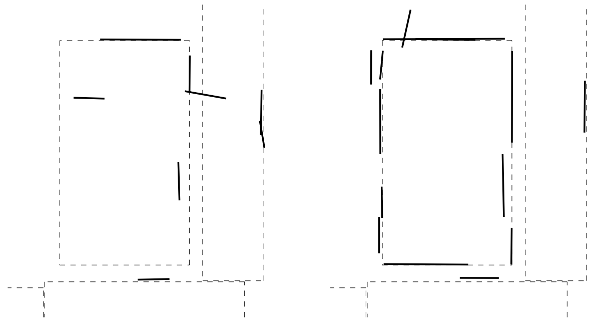

Figure 3: Hand measured model of the living room is shown in dashed lines and the automatically generated ones are in solid lines. Left: Searching for lines in the horizontal plane from one position. Right: Actively moving the platform and using the PTU.

Figure 4: Photos from the living-room when Figure 3 was built.

4.1 Detecting a failure

An effective, way to detect a failure is presented in [13]. By paying attention to weather or not the features that are expected to be seen are actually detected failure situation can be identified. When features predicted to be seen stay undetected for a long enough time, a recovery strategy is initialized. This strategy will fail if; chance has it that measurements match an incorrect predicted feature or the the environment changes so that the visibility of some features are blocked. The detection of can be achieved through use of validation gates as explained for example in [14].

4.2 Recovery from a failure

The recovery strategy is based on using the last verified map, M_v. As long as enough predicted features are detected the verified map is updated, $M_v = M$. When the map no longer can be verified the M_v is left unchanged. Recovery is divided into two steps: localization and restoration. In the localization step the robot localizes itself with respect to M_v. In the restoration step the method used in [13] when changing sub-map is utilized, now with the pose found in the previous localization step acting as a measurement.

5 Multi-sensory maps

As already discussed in Section 1 many different features has been suggested and used in the feature based localization community. The choice of feature depends on the sensors being used. In this paper we will focus on three different features extracted from sonar and laser scanner data. From the laser scanner lines and door features are extracted [15, 16]. The sonar sensor give stable point landmarks extracted using the triangulation based fusion technique [17].

5.1 Sonar based feature detection

To improve the quality of the sonar data triangulation is used. Sonar data typically suffers from, for example, specular reflections and cross talk. To get more accurate information, integration over time is necessary. Keeping in mind the physics of the sonar sensor, the information about the position of the target which reflected the sound is limited to knowing that it is somewhere on an arc (2D assumption about the world). By associating multiple sensor readings to the same target, triangulation can be used to find the true position of the target. The true position is given by the intersection of the circular arcs as illustrated in Figure 5. The method is described in detail in [18].

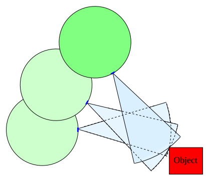

Figure 5: Triangulation based fusion (TBF) of sonar data.

5.2 Laser based feature detection

A typical laser scanner's main characteristics is the high angular resolution that comes with using a laser beam to measure distances. In experiments presented in this paper the SICK laser scanner is used. It has come close to be a standard sensor for modern mobile robots. The high angular resolution makes lines a natural choice. Using an adaptive Hough transform it is relatively easy to extract robust line segments. Door openings can also be extracted. Both types of features are used here. The details of feature extraction and basic pose estimation are outlined in [16].

6 Integrated systems

6.1 ISR system

For evaluation of navigation systems in domestic settings a range of different platforms are used. The techniques presented earlier have been implemented in the Intelligent Service Robot project [19], which is a software system available on a Nomadic Scout, A Nomadic 200 platforms and the Nomadic XR4000 system. The robots are equipped with 16-48 sonars, SICK LMS 200 scanners, and basic odometry. These robots have been used for extensive experiments as outlined in the next section. In total these platforms have driven more than 500 km with the described software system.

6.2 Augmentation of SLAM with geometric constraints

Recently robot platforms for the domestic consumer market have appeared. These robots utilize very limited sensory feedback. This poses a new challenge. Operation in a domestic environment does however at the same time provide addition information. Most houses are constructed according to well defined architectural rules. In many cases such houses have long wall and walls met at well defined angles (typically 90°). In addition furniture is often placed along the walls or away from the walls. Utilizing such information it is possible to feed these constraints into the estimation process. The geometric constraints allow automatic correction of odometric drift which results in a significant improvement in performance as outlined in [13].

6.2.1 The Electrolux example: As an example of a commercial platform that has to rely on SLAM for it operation is the Electrolux vacuum cleaner. The platform is shown in figure 6. The robot uses ultrasonic ranging and odometry for navigation in regular houses. To allow for operation in regular houses a combination of regular SLAM and geometric constraints is utilized.

Figure 6: The Electrolux Trilobite vacuum cleaner, for domestic use, that is commercially available.

7 Experimental Results

7.1 Choice of sub-maps

As has already been stated the standard formulation of the EKF approach to SLAM scales badly to large environments. Figure 7 shows the result of running the algorithm in the lower floor at CAS. The total distance traveled is 388 m. The map consists of 114 lines and the total computational time to build it was 340 s. This was only 12% of the total time of the experiment, but due to the complexity it becomes increasingly difficult to maintain real-time performance when the number of features increases.

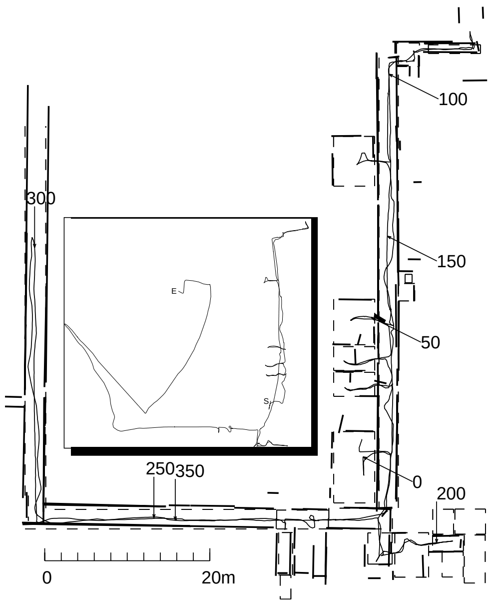

Figure 7: The resulting map when using the standard SLAM algorithm. The position of the robot is marked with an arrow every 50 m. The raw odometric data is also shown.

If sub-maps are used instead the computation time can be reduced. Figures 8 and 9 show the resulting map when 20 by 20 m sub-maps with 2 m overlap are used for two different placements of the sub-maps. The two figures clearly highlights the need to think about how the sub-maps are defined as it effects the quality of the map. The reduction in total computation time is 50% compared to standard SLAM. More importantly, no more than 43 features are processed at the same time which allows for some margin to the real-time threshold.

Picking the right location for the sub-map is difficult when the robot has no knowledge about the world. In Figure 10 the sub-maps are instead defined by rooms and the gateways between the sub-maps are doors. Thus the sub-maps do not have the same size but is instead defined by the environment. In many indoor environments the rooms are small enough for the robot to handle in one sub-map.

7.2 Recovery from errors

To test the ideas for error detection and recovery presented in Section 4 an experiment was performed where a large perturbation was introduced by hand. The system only uses the sonar landmarks presented in Section 5.1. Figure 11 shows the trajectory followed by the robot with the arrow indicating the point at which the perturbation is introduced. Figures 12 shows the pose error made by the SLAM algorithm as a function of time together with the 2σ-bounds. The error is estimated by comparing the pose with the result from a pose tracking algorithm running on the side. The perturbation is clearly visible in the lower sub-figure showing the orientation estimate. At iteration 920 the robot realizes that something is wrong and it successfully performs the recovery step and can continue on with mapping the environment.

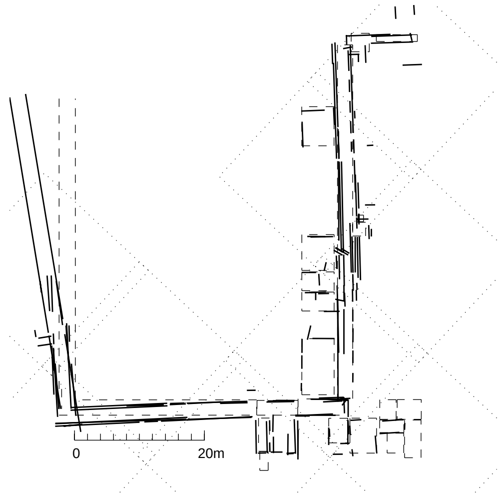

Figure 8: Square sub-maps with poorly picked orientation

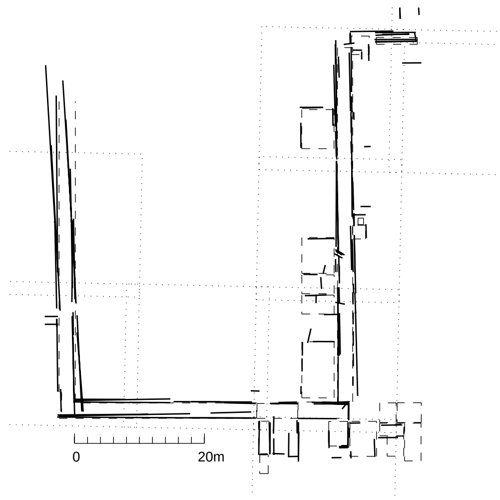

Figure 9: Square sub-maps align with environment

7.3 Utilizing geometric constraints

In many typical indoor environments it is possible to use geometric constraint to simplify the task of localization as many structures have right angles with respect to each other. This is the case not only for the main walls, but we tend to furnish the room by placing book shelves, tables, etc, parallel to the walls. The newly released autonomous vacuum cleaner from Electrolux starts its cleaning cycle by following the walls of a room as well as it can. It will stop the procedure when it estimates to be at the start position again. The vacuum cleaner is equipped with an odometric system as well as short range sonar system for avoiding obstacles and following the wall. The performance of current system is limited by the drift accumulated by the odometry under normal conditions and suffers badly from large perturbations that happens when the robot drives over things that are on the floor. By utilizing geometric constraints the robot can reduce the uncertainty substantially. Figure 13 shows the result with and without using the geomet-

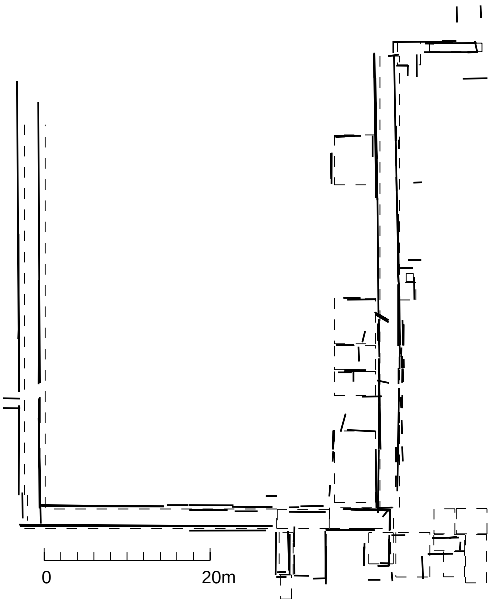

Figure 10: Room based square sub-maps

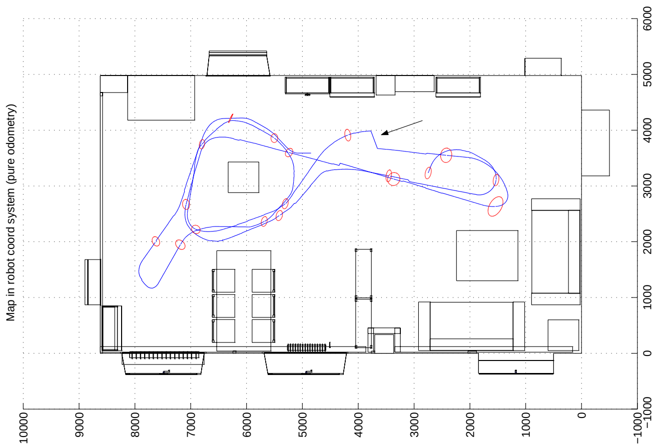

Figure 11: A perturbation is introduced at the arrow.

ric constraints for wall following.

8 Summary and discussion

In this paper the basic use of SLAM has been outlined and it has been outlined how the method may utilize hybrid representations to provide an overall system that has scalable complexity. In addition it has been described how the method may detect and recover from error situations. Finally the use of geometric constraints may be used for automatic correction of odometric slippage so as to limited the updating rate of the SLAM method. The presented methods have been extensively tested on in-door environment on four different platforms.

For large scale SLAM there is still a need for methods for automatic generation of topological maps that can be used for overall planning and embedding of the local geometric maps. Future research will thus emphasize methods for automatic hybrid mapping to ensure bounded complexity for operation in truly large scale environments.

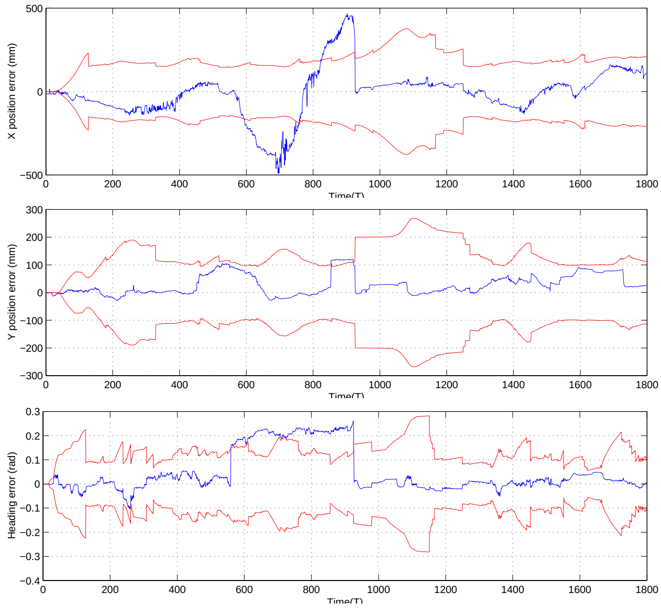

Figure 12: Error and 2σ-bound for *x*, *y* and orientation. The perturbation is clearly visible in the orientation plot.

References

[1] H. Durrant-Whyte, "Where am i," *Industrial Robot*, vol. 21, pp. 11–16, 1994.

[2] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'85)*, pp. 116–121, IEEE, 1985.

[3] J. J. Leonard, H. F. Durran-Whyte, and I. J. Cox, "Dynamic

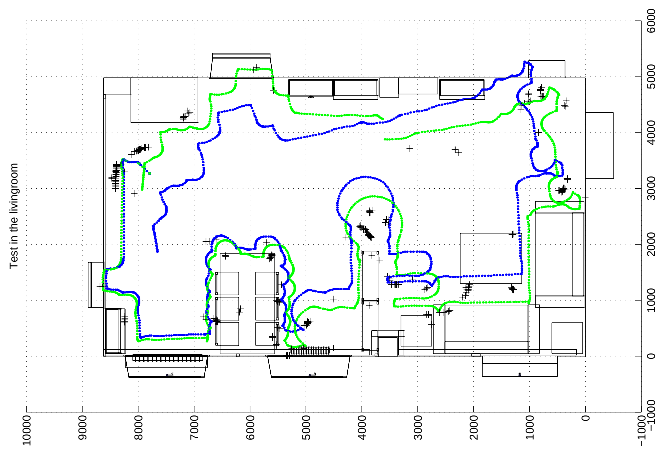

Figure 13: Using geometric constraints when doing wall following

map building for an autonomous mobile robot," *The International Journal of Robotics Research*, vol. 11, no. 4, pp. 286–298, 1992.

[4] R. Chatila, "Mobile robot navigation: Space modeling and decisional processes," in *Proc. of 3rd International Sysmposium on Robotics Research*, (Gouvieux, France), pp. 373–378, Oct. 7-11, 1985.

[5] M. Drumheller, "Mobile robot localization using sonar," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 325–331, Feb. 1987.

[6] K. Arras and N. Tomatis, "Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision," in *Proc. of the 3rd European Workshop on Advanced Mobile Robots (Eurobot'99)*, (Zürich, Switzerland), pp. 177–185, Sept. 6-9, 1999.

[7] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. of the International Workshop on Intelligent Robots and Systems*, vol. 3, (Osaka, Japan), pp. 1442–1447, IEEE, Nov. 3-5, 1991.

[8] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *4th International Symposium on Robotics Research*, 1987.

[9] J. J. Leonard and H. J. S. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *Robotics Research* (J. J. Hollerbach and D. Kodichek, eds.), Berlin: Springer Verlag, March 2000.

[10] H. J. S. Feder, *Simultaneous Stochastic Mapping and Localization*. PhD thesis, MIT, June 1999.

[11] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization by entropy minimization," in *Proc. of the 2nd Euromicro Workshop on Advanced Mobile Robots*, IEEE/CS, 1997.

[12] M. Seiz, P. Jensfelt, and H. I. Christensen, "Active exploration for feature based global localization," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)"*, vol. 1, pp. 281–287, 2000.

[13] G. Zunino, *Simultaneous Localization and Mapping for Navigation in Realistic Environments*. Lic.tech. thesis, Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, February 2002.

[14] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. New York, NY.: Academic Press, 1987.

[15] P. Jensfelt and H. I. Christensen, "Pose tracking using laser scanning and minimalistic environmental models," *IEEE Trans. on Robotics and Automation*, vol. 17, pp. 138–147, April 2001.

[16] P. Jensfelt, *Approaches to Mobile Robot Localisation in Indoor Environments*. PhD thesis, Electrical Enginering and Computer Science, Royal Institute of Technology, S3, KTH, SE-10044 Stockholm, June 2001. ISBN 91-7283-135-9.

[17] O. Wijk, *Triangulation Based Fusion of SOnar Data with Application in Mobile Robot Mapping and Localisation*. PhD thesis, Centre for Autonomous Systems, Royal Institute of Technology (KTH), 10044 Stockholm, Sweden, April 2001.

[18] O. Wijk and H. I. Christensen, "Triangulation-based fusion of sonar data with application in robot pose tracking," *IEEE Trans on Robotics and Automation*, vol. 16, pp. 740–752, Dec. 2000.

[19] M. Andersson, A. Orebäck, M. Lindström, and H. Christensen, *Intelligent Sensor Based Robots*, vol. 1724 of *Lecture Notes in Artificial Intelligence*, ch. ISR: An Intelligent Service Robot, pp. 291–314. Heidelberg: Springer Verlag, October 1999.