A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping [1]

Sebastian Thrun[*] Wolfram Burgard[†] Dieter Fox[‡]

Mohammad Faisal
Jacobs University Bremen gGmbH
School of Engineering and Science
Campus Ring 1, D-28759 Bremen, Germany
m.faisal@jacobs-university.de

Abstract: This paper presents a novel approach, which combines the iterative and expectation maximization approaches, into an incremental method for concurrent mapping and localization for mobile robots. The approach uses 2D laser scanners to generate simple 3D maps of the environment. The simplification of the 3D maps is done using multi-resolution approach from computer graphics literature. The approach uses scan matching for mapping and a sample based probabilistic method for localization, which enables it to work with multiple robots, and in large cyclic environments. The approach is extremely robust, and can create accurate maps of the terrain without the use of odometry, in an unknown environment. The results show the robustness of the approach in numerous situations.

1 Introduction

The word "autonomous" comes from the Greek words "auto" and "nomos", which mean, "self" and "law" respectively. Hence the term autonomous means something which is self governing, self controlling. The term autonomous have been used a lot recently in conjunction with robotics. The idea of having a robot which is completely autonomous is the current research goal, and much work has been done in the research area. The biggest problem to solve is that of *simultaneous localization and mapping* (**SLAM**) or *concurrent mapping and localization* (CMAL). Before I define the problem, it is important to explain these terms. **Localization** means, the act of keeping track of oneself in an environment. The SLAM/CMAL problem is defined as follows: *Given an unknown environment, and an unknown start position, there should be an approach to create a map of the environment, while keeping track of the movement of the robot.* Although this seem like a simple problem, it is quite difficult due to the "chicken and the egg" nature of it. Work done by Moraves and Elfes demonstrate how maps can be created when the location of the robot is known [2]. These tests involved taking sonar readings of an object from number of known poses (a **pose** is an x, y, theta coordinate of the robot relative to a certain plain), and triangulating the position of the object. There has also been a lot of work done in regards to localization in a known environment. This involves the idea of map matching, among others, where a scan is matched to the map in the local frame of the robot [3].

[*] Carnegie Mellon University, Pittsburgh, PA, USA
[†] University of Freiburg, Freiburg, Germany
[‡] Carnegie Mellon University, Pittsburgh, PA, USA

There has been some progress made recently to combine the both areas. Most of these approaches iteratively localize and map, using each new sensor scan the robot receives. It builds the maps incrementally [4–7]. The basic idea is quite simple. The approach takes a scan, and creates a map out of it, then it uses the previous scan, and the action performed, to move in the previous scan. At this point, it takes a new scan and includes that in the overall map. This iterative process makes the approach quite fast, and can be used in real-time applications. However, since the error in each step is accumulated, the overall error after a number of steps, in a large cyclic environment, can be quite large. **Cyclic** environment are ones which contains a lot of cycles and loops. This is due to the fact that at each step, we have a map which cannot be modified; hence there is no concept of backwards correction (4.2).

To solve this problem, a number of Expectation Maximization approaches have also been used recently [8,9]. Expectation Maximization (**EM**) is an area, which uses probabilistic methods to find the most likely position of something in a certain environment. The simplest example is that of a best fit line, for a selection of points. It works by searching through all the possible points at the same time. In the example of the best fit line, this is 179 degrees of gradient of the line. This way, it considers all possible positions in all previous scans, simultaneously, using an iterative refinement procedure to narrow down to most likely position in the global map. These approaches have shown good results in large cyclic environments, but due to the large number of possibilities, and large run-time; these approaches are normally used in batch, in a non-real-time application.

Getting the pose is only the first part in creating a 3D Map. The second step is actually relating the sensor data with the pose. Most work which has been done so far focuses mostly on 2D mapping, using a single robot. Our focus is on 3D mapping through the use of multiple robots.

Our approach is to combine the both types of algorithms described above (EM and incremental), in a way such that the advantages of both are utilized. This includes fast real-time application, and robustness in large, cyclic environments. This is realized in a novel approach by combining the idea of posterior estimate with that of incremental map construction using maximum likelihood estimators. The approach turns out to be very robust in large cyclic environments, which still being able to run real-time on a low-end computer. It also allows data collection through multiple robots by having them localize relative to each other. The use of two laser scanners, (one horizontal and one vertical) also allows us to create a 3d map, where a multi-resolution algorithm is used to reduce the complexity of the map. Results showed that the approach is very robust, even in absence of odometry data.

The structure of the paper is as follows: In section 2 we define the problem we are aiming to solve. In section 3 we describe the existing state of the art approaches towards solving the problem. In section 4, we explain the four main parts of our approach. This is followed by results in section 5, ending with a summary in section 6.

2 Problem Definition

The problem of concurrent mapping and localization can be treated as a search where the goal is to find most likely map for given data. A map is a collection of scans and their poses [10,11]. The term pose refers to a tuple $< x, y, \theta >$. Here x and y refers to a location relative to some hypothetical coordinate system, and the θ refers to the orientation of the scan. Let m be a map. Then at time t, the map is written as

$$m_t = \{\langle o_\tau, \hat{s}_\tau \rangle\} \tag{1}$$

where $\tau = 0, ..., t$ are time intervals, o_τ denotes a laser scan and $\hat{s}_\tau$ is the pose of the scan. As mentioned above, our goal is to find the most likely map given the data d_t. This is shown below:

$$d_t = \{s_0, a_0, s_1, a_1, ..., s_t\} \tag{2}$$

$$\arg\max_m P(m \mid d_t) \tag{3}$$

Here d_t is a sequence of laser scans (s_τ) and corresponding odometry readings (a_τ). We assume here that observations and odometry readings are alternated.

3 Existing Approaches

3.1 Normal Incremental Mapping

The conventional incremental approach is very popular in the literature. This is considered a baseline approach towards solving the concurrent localization and mapping problem. It is incremental but is unable to revise the map backwards in time, and therefore accumulates errors. Therefore, loop closing cannot be utilized to reduce the error. **Loop Closing** is when a robot goes around and creates a cycle (closes a loop) by passing through the same location twice. When a loop closing is detected, the following problems are detected in this approach:

1. Since the pose errors accumulate, they can grow very large. Then when closing the loop in a cyclic environment, search algorithms like gradient descent can fail to find the optimal solution.
2. When a loop is closed in a cyclic environment, past poses may need to be revised to distribute the error, and generate a consistent map. This approach is incapable of doing this.

The idea is very simple, and therefore, very popular: Given a scan and an odometry reading, determine the most likely pose for the scan. Then append the pose and the scan to the map. Freeze it once and forget about it. This is represented below:

$$\hat{s}_t = \arg\max_{s_t} P(s_t \mid o_t, a_{t-1}, \hat{s}_{t-1}) \tag{4}$$

The pose is usually searched through hill climbing with gradient descent. This result is then added to the map as below:

$$m_{t+1} = m_t \cup \{\langle o_t, \hat{s}_t \rangle\} \tag{5}$$

As it can be seen, only a single guess is maintained as to where the robot is at each scan, instead of a full distribution. Furthermore, since the past estimates are never revised, this approach becomes very brittle in large cyclic environments. An example is shown in figure 1. Here it can be seen that when the robot creates a loop, a noticeable error leads to mismatching of the wall. This is because in the simple incremental approach, error accumulates over time.

An important thing to note here is that these restrictions do not apply to EM family of mapping algorithms [8, 9].

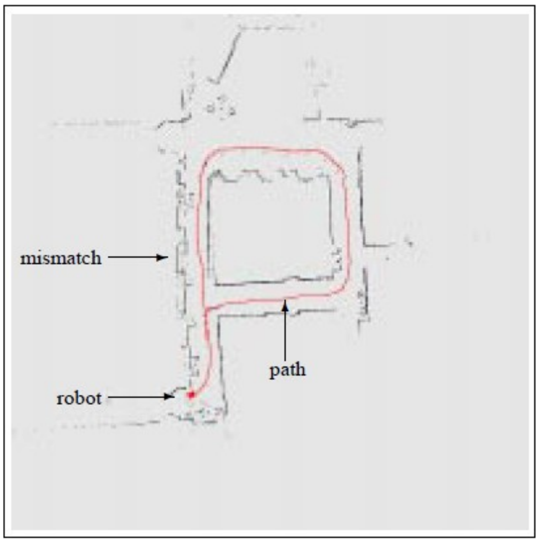

Figure 1: Map created through a simple incremental approach. Since error accumulates, when a loop is closed, there is noticeable error.

3.2 Likelihood Function

As shown in equation 3, our goal is to find most suitable pose for the given data and associated map $P(m \mid d_t)$. The likelihood function has been derived in [9] as follows:

$$P(m \mid d_t) = \eta P(m) \int ... \int \prod_{\tau=0}^{t} P(o_{tau} \mid m, s_\tau) \times \prod_{\tau=0}^{t-1} P(s_{\tau+1} \mid a_\tau, s_\tau) ds_1 ... ds_t \tag{6}$$

where n is a normalizer and $P(m)$ is prior over the maps, both of which can be omitted since we are only concerned with the most suitable pose. Thus the equation simplifies to a function with two terms, the motion model, $P(s_{\tau+1} \mid a_\tau, s_\tau)$ and the perceptual model, $P(o_\tau \mid m_\tau, s_\tau)$. Since neither of these models depends on time, we can assume them to be stationary. This gives us $P(s \mid a, \acute{s})$ for the motion model and $P(o \mid m, s)$ for the perceptual model.

An important aspect of these simplified models is that they are differentiable. This leads to the following gradients for efficiently searching for the most likely pose of the robot, given its sensor measurements.

$$\frac{\partial P(s \mid a, \acute{s})}{\partial s} \qquad \frac{\partial P(o \mid m, s)}{\partial s} \tag{7}$$

These gradient computations are carried out highly efficiently in our implementation, enabling more than 1000 gradient computations per second on a low-end PC. This fact is used in our approach to search for the most optimal pose estimate later in hill climbing.

The probabilistic motion model we use in this paper is shown in figure 2. Assuming we know the starting position of the robot, $\acute{s}$ and the executed action a, the figure shows the probability of being at pose s. The banana shape is due to the fact that the robot motion can have noise in the rotational and the translational component. It can be seen that the probability distribution for the simple motion is more focused, compared to the one with the

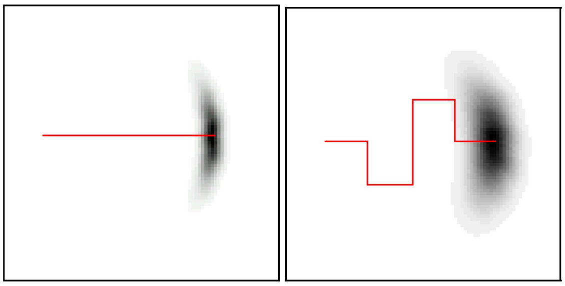

Figure 2: The probability distribution $P(s \mid a, \acute{s})$ for the robot's posterior pose s after moving distance a beginning at location $\acute{s}$ for two different trajectories.

more complicated motion. This model is based on simple kinematic equations as demonstrated in [11].

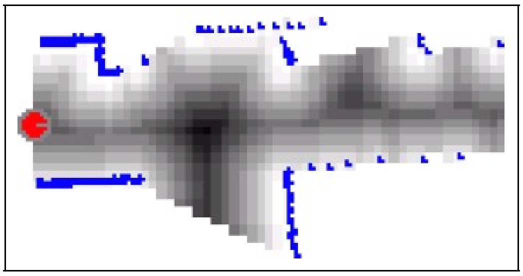

Figure 3: The probability distribution of an object 180 degrees in front of a robot (red dot). The darker region suggest smaller likelihood of an object being present.

The probabilistic perceptual model is shown in figure 3. This has been inherited from literature on scan matching and projection filtering [10, 11]. The assumption here is that a received sensor scan will be unlikely to show an object at a place which has been indicated as free space in a previous scan. This likelihood is dependent on the distance traveled. The larger the distance traveled, the lower the likelihood. In the figure, this is shown by areas marked as gray. The darker the color, the smaller the likelihood of an object being detected there in subsequent scans. The robot is shown as the red dot. Another thing to notice here are that the areas which are occluded (cannot be seen from the sensor) and marked white, and are not part of any subsequent calculations. This is often detected using ray-tracing. An example of how this sensor model works is shown in figure 12 in appendix.

We can see that maximization of equation 6 results in the most likely output map. Since the likelihood function cannot be maximized incrementally, it cannot be used in real-time. In addition, it replies on the sensor measurements to get information about the previous robot locations, which can then be revised to get a global optimal map. As mentioned in [10, 11], when a loop is closed in a large cyclic environment, the error could be very large. In such cases, when a loop is closed, an optimal map requires correction of the map backwards in time. This is the main reason such approaches in the past, such as ones in [8, 9] are off-line algorithms which may sometimes take multiple hours to compute the most likely map.

4 Our Approach

Our approach has four main parts. The first is the calculation of the pose, the second is the ability to optimize globally once a loop is detected, the third is the ability to use multiple robots to map and the fourth is the ability to create 3d maps. This section explains these four parts.

4.1 Pose Calculation

In equation 4, the approach computes the maximum likelihood pose. Our approach, instead, computes the full **posterior** over robot poses. This approach is based on the literature on probabilistic mapping and Markov localization as shown in [8, 9, 12, 13]. *Posterior* is a probability distribution over poses conditioned on past sensor data and is represented as below:

$$Bel(s_t) = P(s_t \mid d_t, m_{t-1}) \tag{8}$$

The *Bel* in equation 9 stands for robots belief as to where it is. Algorithms which maintain a posterior estimation, instead of a single maximum likelihood guess, are more robust as they offer versatility to improve previous estimates based on new data, to create an globally optimal solution. The algorithm we are using is based on the Markov localization algorithm [12].
Our approach, like Markov localization, is incremental. At time $t = 0$, the belief $Bel(s_0)$ is centered on origin. Suppose at time t, we know the previous belief $Bel(s_{t-1})$, which is distribution over poses s_{t-1} at time $t - 1$, and we execute action a_{t-1} and observe o_t. Then the new belief is:

$$Bel(s_t) = \eta P(o_t \mid s_t, m_{t-1}) \int P(s_t \mid a_{t-1}, s_{t-1}) Bel(s_{t-1}) ds_{t-1} \tag{9}$$

where η is a normalizer and m_{t-1} is the best available map. The derivation is shown in the appendix (13). Once this *posterior* is computed, the new pose is estimated using the following equation:

$$\bar{s}_t = \arg\max_{s_t} Bel(s_t) \tag{10}$$

From this, we get the new map as follow:

$$m_{t+1} = m_t \cup \{\langle o_t, \bar{s}_t \rangle\} \tag{11}$$

This is similar to the incremental approach as in equation 4 and 5 but the main difference lies in how $\hat{s}$ and $\bar{s}$ are calculated. Whereas $\hat{s}$ is calculated using the most recent sensor scan and the pose, $\bar{s}$ is calculated using the entire posterior $Bel(s_t)$. However, to narrow down the search space when closing a loop, a diameter of uncertainty is modeled. This is similar to the motion model explained earlier. If the loop is large, then the margin of uncertainly is large as well.

Our approach also makes use of samples to approximate the posterior, instead of using analog values. In relation to posterior estimation, it is called a *particle filter* [14] and the approach is equivalent to *Monte Carlo localization.* This is represented in figure 4. The shape is similar to the banana shape as observed in figure 2. Each of the sample sets is an approximation of the densities (darkness in figure 2). This results in a convergence rate of $\frac{1}{\sqrt{N}}$ for approximating an arbitrary posterior [15]. Sampling has been used in numerous areas of research, from Rubin's importance sampler [16] to tracking in computer vision [17] to mobile robot localization [13, 18]. One factor for such wide-spread usage is the ease of implementation of a particle filter [13]. An example of this is shown in figure 13 in appendix.

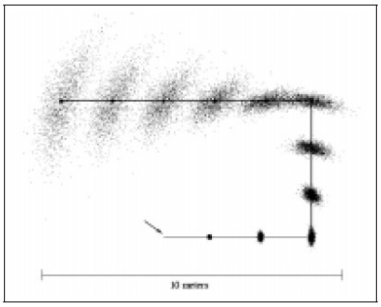

Figure 4: Sample based approximation for posterior $Bel(s)$. Density is represented by a set of samples, weighted by numerical importance.

As mentioned in section 3.1, since a single starting pose is used for the hill-climbing search, it might fail to produce a globally optimal map, especially when a loop is closed. The sample-based representation described above, and being used in our approach directly facilitates optimization in equation 10 to get an optimal pose estimate. The approach uses each sample as a starting point for gradient descent, and then calculates a likelihood function using it. If the samples are spaced densely, it is likely that a global maximum can be found and a globally optimal map can be generated.

4.2 Backward Correction

When the robot is moving in an environment without loops, the pose calculated through both equation 10 and 4 are the same. These values defer when a loop is detected, hence our backward correction is dependent on the following equation:

$$\Delta s_t = \bar{s}_t - \hat{s}_t \tag{12}$$

When $\Delta s_t \neq 0$, the posterior estimate is different, hence poses have to be revised backwards in time. Our approach handles this in three steps:

1. Size of the loop is determined by checking the scan in the map which leads to the adjustment.
2. The error Δs_t is distributed proportionally among all poses in the loop. This places the intermediate poses in a good starting match for subsequent gradient search.
3. Gradient descent is applied iteratively for all the poses which form the loop until the map is globally optimal under the constraint created from the loop detection.

These three steps are an efficient approximation for maximum likelihood estimate for the entire loop, which can be implemented very fast.

4.3 Multi-Robot Mapping

The posterior estimation component of our approach makes it easy to generate maps with multiple robots. We assume that although the initial pose of the robots relative to each other is unknown, each robot starts within the map of a specific robot called team leader. Each robot then localizes itself in the map of the team leader, to generate a single map. It then becomes a simpler problem of localization in a known map. This is illustrated in figure 5. In the start, the initial samples of the robot are uniformly distributed in the map of the team leader. As the robot moves, the uncertainly decreases until the robot finds it pose in b. Further movement increases the area of uncertainty.

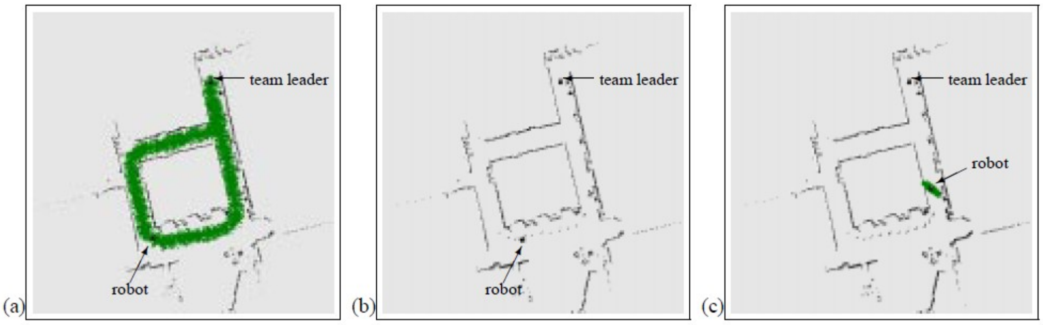

Figure 5: a) Uniform distribution in the existing map to show initial uncertainly. b) The robot finds it pose relative to existing map. c) Increasing area of uncertainty as the robot moves.

4.4 3D Mapping

One important goal of our research is to be able to generate accurate 3D Maps. Using two laser range finders, as shown in figure 10b in appendix, 3D maps of the environment can be generated. The forward looking laser scanner is used for localization and mapping, whereas the upward looking one is used to generate 3D map of the environment. Given the accurate pose estimate, creating the 3D map is simple to do by connecting nearby laser scans with each other; however this leads to large noise, and overly complex output map.

In order to simplify the 3D Map, our approach utilizes a number of techniques. The first is a constraint is used to remove outliers based on distance. If two scans are separated by more than twice the distance, the robot moved between those scans, these scans are not considered. Furthermore, we use a simplification algorithm which has been developed to simplify polygonal models for real-time rendering in computer graphics [19, 20]. It does this by iteratively simplifying multi-polygon surface models by fusing those which look similar when rendered. This results in a simple model in terms of complexity, which is similar to the original in terms of accuracy and looks, which can be used in real-time.

5 Results

Before the results are shown, it is important to clarify how the data is used in the creation of the maps and in localization. To reduce the complexity, scans are only added to the map if the robot has moved 2 meters. This reduces the size of the map from thousands of scans to hundreds of scans. However, to keep the localization accurate, all scans are used in it. To check the robustness of the approach, random errors were introduced into the odometry data.

5.1 Resources and Time

One important aspect of the research was to have an approach which is real-time. As mentioned in section 4.1, the sampling allows for a very large convergence rate, and that combined with the simple gradient decent calculation of likelihood function, enable very fast calculation of the pose. Furthermore, the backwards tracking is also extremely fast, with can be carried out between two sensor measurements for all experiments in this paper. Also, the simplification of the 3D model of the map leads to it being real-time. Overall, all the results shown in this paper have been obtained in real-time on a low-end computer.

5.2 Loop Closing

Figure 6 is a good illustration of how well the backward correction works. The robot moves and the uncertainty around it increases in 6a and 6b. The posterior belief is represented by the dots centered around the maximum likelihood pose. When the loop is closed in 6b, the error is significant and can be seen in front of the robot by the misaligned walls. However, when a loop is closed in 6c, our approach identifies the error, and corrects past beliefs and reduced the robots uncertainty accordingly. The result is an accurate map as seen in 6c, where the error from the cycle has been eliminated.

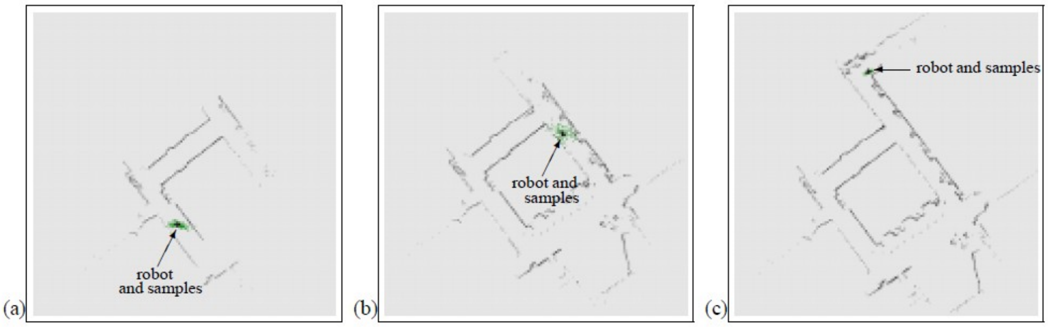

Figure 6: a) and b) shows increasing uncertainty around the robot as it moves. c) shows reduced uncertainty once a loop is closed.

5.3 Robustness

To test the robustness of the approach, we attempted to build the same map in the absence of odometry data. The raw data without odometry is shown on the left in figure 7. The corrected map is shown on the right in figure 7. It can be seen visually that this looks very similar to the one generated in the previous section. One important thing to keep in mind is that the approach only works in environment with sufficient variations and will fail in the absence of odometry data in a featureless environment. Nevertheless, the map illustrates the robustness of the approach.

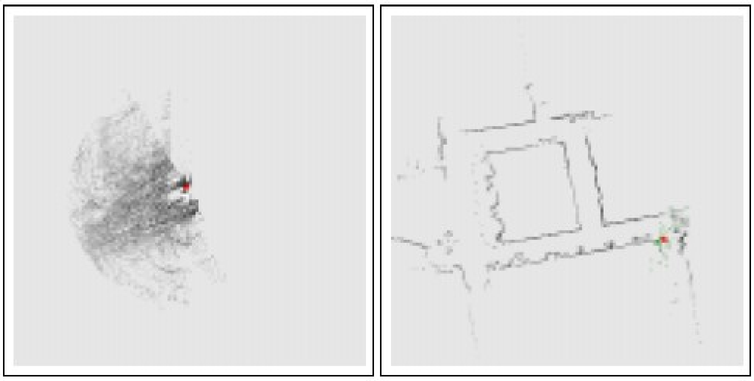

Figure 7: Left: No odometry. Right: Output map

To further demonstrate the robustness, we used the robot shown in appendix in 10c. This robot uses tracks and has errors often as large as 100%, depending on the conditions of the floor. This could lead to very bad results as seen on left in figure 8. The right image shows the output map with the robot's trajectory. This map is very interesting as it represents the worst results from all the experiments in this paper. Some of the walls are not aligned, and are rotated by up to 2 degrees. Nevertheless, given the original raw data, and the fact that our approach is not constraint by orthogonal walls, the results are fairly accurate and sufficient for navigation. Furthermore, once the environment is explored more, loops could be created, improving the global map.

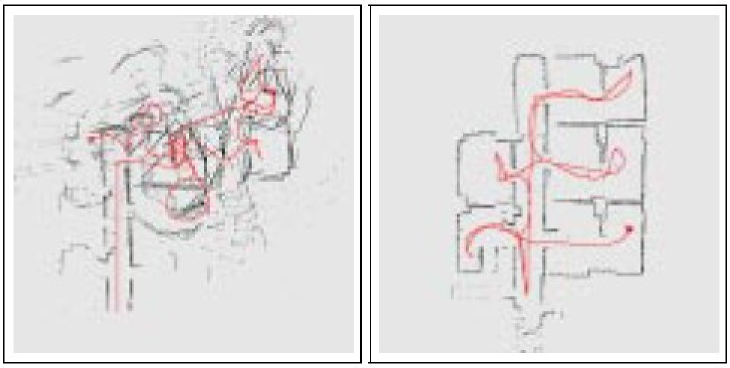

Figure 8: Left: Bad odometry. Right: Output map

5.4 Multi-Robot Mapping

Figure 5 demonstrates the results of mapping through multiple robots. In a, the robot initializes itself in the map of the team leader robot, with uniform distribution across the map. After some motion, the posterior is focused on a single location in b, and the incoming sensor data is used to further build the map. Part c shows the situation after a few seconds with robot progressing through the map of the team leader. The robot still knows its location but uncertainty is increasing with the motion.

5.5 3D Mapping

Figure 9, and in appendix figure 11, show results for the 3D mapping. Figure 9 shows a short corridor section. The corridor can be seen, and sections of roof which was detected through

the open door is also visible.
Figure 11 shows a larger section, of about 60 meters long. The rendering algorithm is a standard virtual reality tool (VR-web) which enables the user to remotely inspect the building by ”flying through the map”. The top row is generated from raw laser data and it contains 82,899 polygons. In contrast, the simplified polygonal model is on the bottom row, which contains only 8,289 polygons. Visually, both look identical and have similar level of accuracy; however, the lower one is 10 % of the complexity as the one created from raw data.

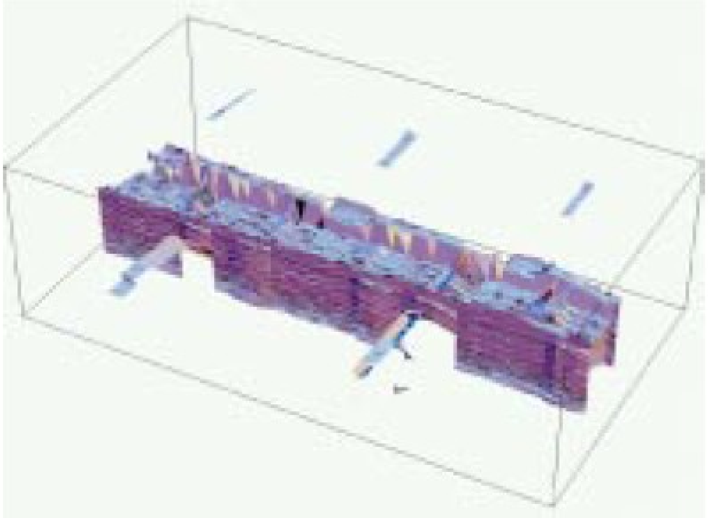

Figure 9: 3D Map

6 Discussion

The approach presented in this paper is new, real-time, robust, on-line approach towards concurrent mapping and localization in indoor environments. The approach combines advantages from incremental mapping with advantages from posterior estimation and backwards correction to produce a fast and robust algorithm which can be extended to multiple robots and 3D mapping. The end result of our approach is simple, compact real-time rendering of the 3D model of the environment. The experimental results show the robustness of the approach, even in cases where odometry will not provided. The results were highly accurate maps, with the worst test case having misalignment of about 2 degree with some walls. All of this was done in real-time on a low-end PC. This is a big improvement over existing EM approaches towards concurrent mapping and localization, which sometimes multiple hours to complete. Our approach uses implementation of posterior estimate and back correction which is extremely fast, resulting in processing of large cyclic environments in real-time. Just by adding a few more constraint, such as checking for orthogonal surfaces, the results of our approach could be improves significantly. Nevertheless, our approach surpasses all other approaches for concurrent mapping and localization, in terms of speed, accuracy and robustness.[1] [2]

References

[1] S. Thrun, W. Burgard, and D. Fox, “A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping,” pp. 321–328, 2000.

[1] Associated presentation at http://www.powershow.com/view/2ac219-MjJiN/Machine_Vision_Seminar

[2] Associated videos at http://robots.stanford.edu/videos.html

[2] H. Moravec and A. E. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, pp. 116 – 121, March 1985.

[3] J. Borenstein, H. R. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques.* Natick, MA, USA: A. K. Peters, Ltd., 1996.

[4] J. Leonard, H. Durrant-Whyte, and I. Cox, “Dynamic map building for autonomous mobile robot,” in *Intelligent Robots and Systems ’90. ’Towards a New Frontier of Applications’, Proceedings. IROS ’90. IEEE International Workshop on*, pp. 89 –96 vol.1, July 1990.

[5] W. Rencken, “Concurrent localisation and map building for mobile robots using ultrasonic sensors,” in *Intelligent Robots and Systems ’93, IROS ’93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 3, pp. 2192 –2197 vol.3, July 1993.

[6] B. Yamauchi and R. Beer, “Spatial learning for navigation in dynamic environments,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 26, pp. 496 –505, June 1996.

[7] B. Yamauchi, P. Langley, A. C. Schultz, J. Grefenstette, and W. Adams, “Magellan: An integrated adaptive architecture for mobile robotics,” 1998.

[8] H. Shatkay, “Learning models for robot navigation,” 1998.

[9] S. Thrun, W. Burgard, and D. Fox, “A probabilistic approach to concurrent mapping and localization for mobile robots,” *Auton. Robots*, vol. 5, pp. 253–271, July 1998.

[10] J.-S. Gutmann and C. Schlegel, “Amos: comparison of scan matching approaches for self-localization in indoor environments,” in *Advanced Mobile Robot, 1996., Proceedings of the First Euromicro Workshop on*, pp. 61 –67, Oct. 1996.

[11] F. Lu and E. Milios, “Robot pose estimation in unknown environments by matching 2d range scans,” *J. Intell. Robotics Syst.*, vol. 18, pp. 249–275, March 1997.

[12] R. Simmons and S. Koenig, “Probabilistic robot navigation in partially observable environments,” in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, (San Francisco, CA, USA), pp. 1080–1087, Morgan Kaufmann Publishers Inc., 1995.

[13] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte carlo localization: Efficient position estimation for mobile robots,” in *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI’99).*, July 1999.

[14] M. K. Pitt and N. Shephard, “Filtering via Simulation: Auxiliary Particle Filters,” *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[15] M. A. Tanner, *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions.* Springer Series in Statistics, Springer Verlag, 3rd ed., 1996.

[16] D. B. Rubin, “Using the SIR algorithm to simulate posterior distributions,” in *Bayesian Statistics 3* (M. H. Bernardo, K. M. Degroot, D. V. Lindley, and A. F. M. Smith, eds.), Oxford University Press, 1988.

[17] M. Isard and A. Blake, “Condensation – conditional density propagation for visual tracking,” 1998.

[18] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, “Using the condensation algorithm for robust, vision-based mobile robot localization,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999.

[19] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH ’97, (New York, NY, USA), pp. 209–216, ACM Press/Addison-Wesley Publishing Co., 1997.

[20] M. Garland and P. S. Heckbert, “Simplifying surfaces with color and texture using quadric error metrics,” in *Proceedings of the conference on Visualization ’98*, VIS ’98, (Los Alamitos, CA, USA), pp. 263–269, IEEE Computer Society Press, 1998.

A Appendix

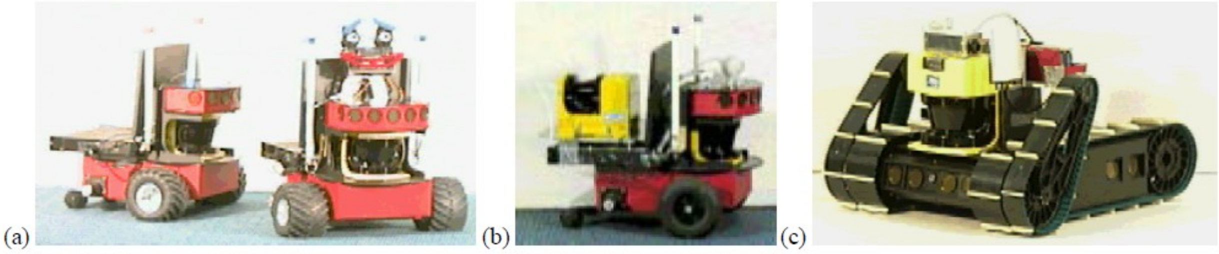

Figure 10: Robots manufactured by RWI/ISR : a) Pioneer robots used for multi-robot mapping. b) Pioneer robot with 2 laser range finders used for 3D mapping. c) Urban robot for indoor and outdoor exploration.

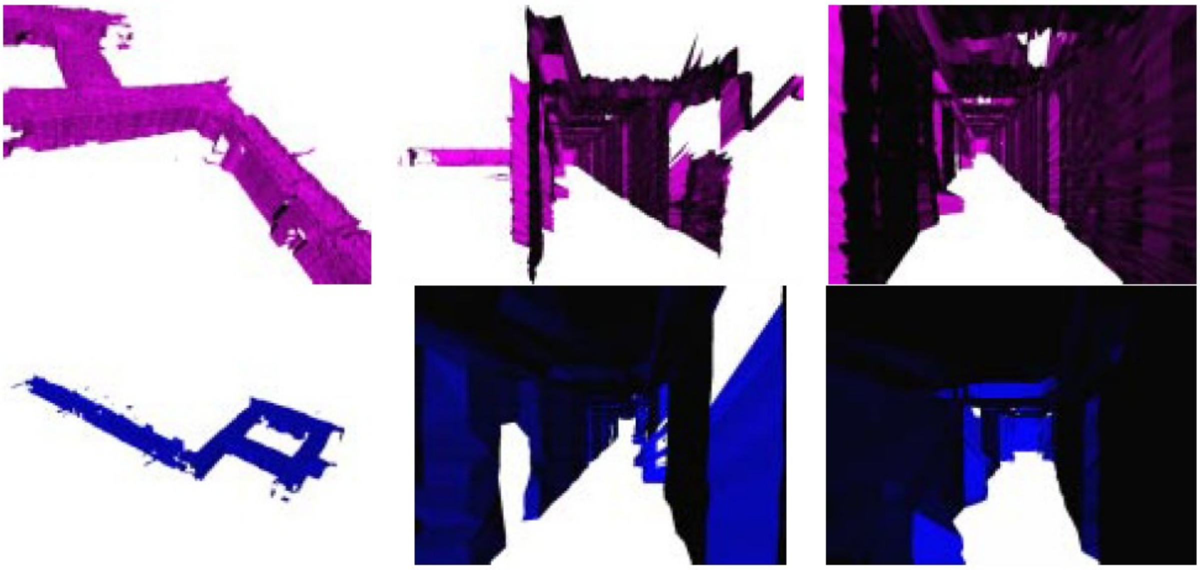

Figure 11: Views of the 3D map, for high resolution model (top row) and low resolution model (bottom row)

From the common Markov localization approach, as shown in [12,13], assuming a static map, we can derive the following equations:

$$
\begin{aligned}
Bel(s_t) &= P(s_t \mid d_t, m_{t-1}) && (13)\\
&= P(s_t \mid d_{t-1}, a_{t-1}, o_t, m_{t-1}) && (14)\\
&= \eta P(o_t \mid s_t, d_{t-1}, a_{t-1}, m_{t-1}) P(s_t \mid d_{t-1}, a_{t-1}, m_{t-1}) && (15)\\
&= \eta P(o_t \mid s_t, m_{t-1}) \int P(s_t \mid d_{t-1}, a_{t-1}, s_{t-1}, m_{t-1}) && (16)\\
&\qquad P(s_{t-1} \mid d_{t-1}, a_{t-1}, m_{t-1}) ds_{t-1} &&\\
&= \eta P(o_t \mid s_t, m_{t-1}) \int P(s_t \mid a_{t-1}, s_{t-1}) Bel(s_{t-1}) ds_{t-1} && (17)
\end{aligned}
$$

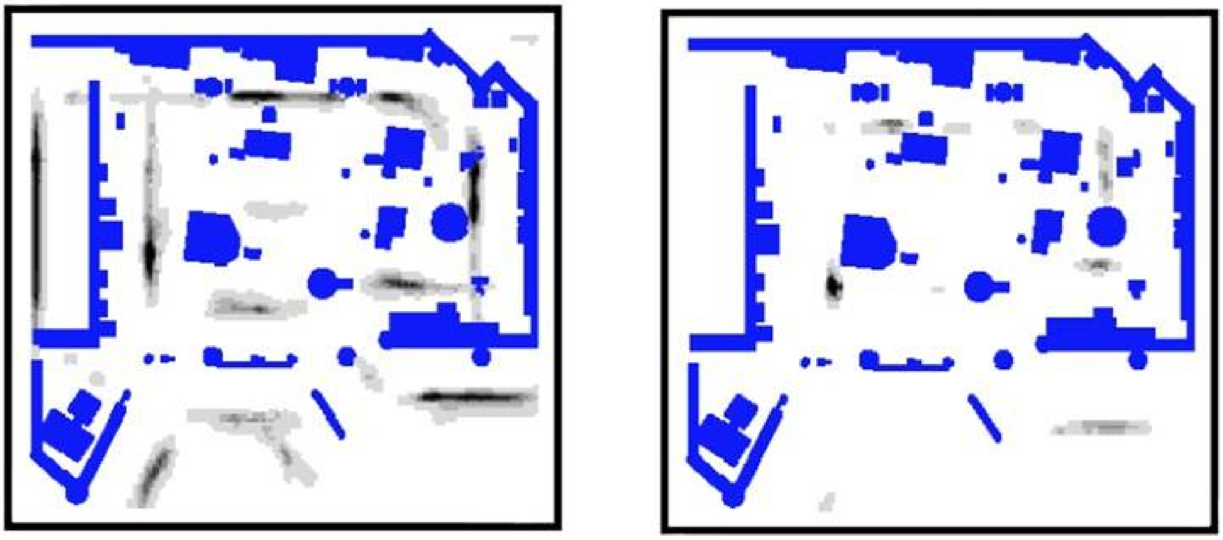

Figure 12: Left map shows possible places on the map where the robot could be. The right map shows the situation after a few minutes. The search area is now more focused around the actual position of the robot.

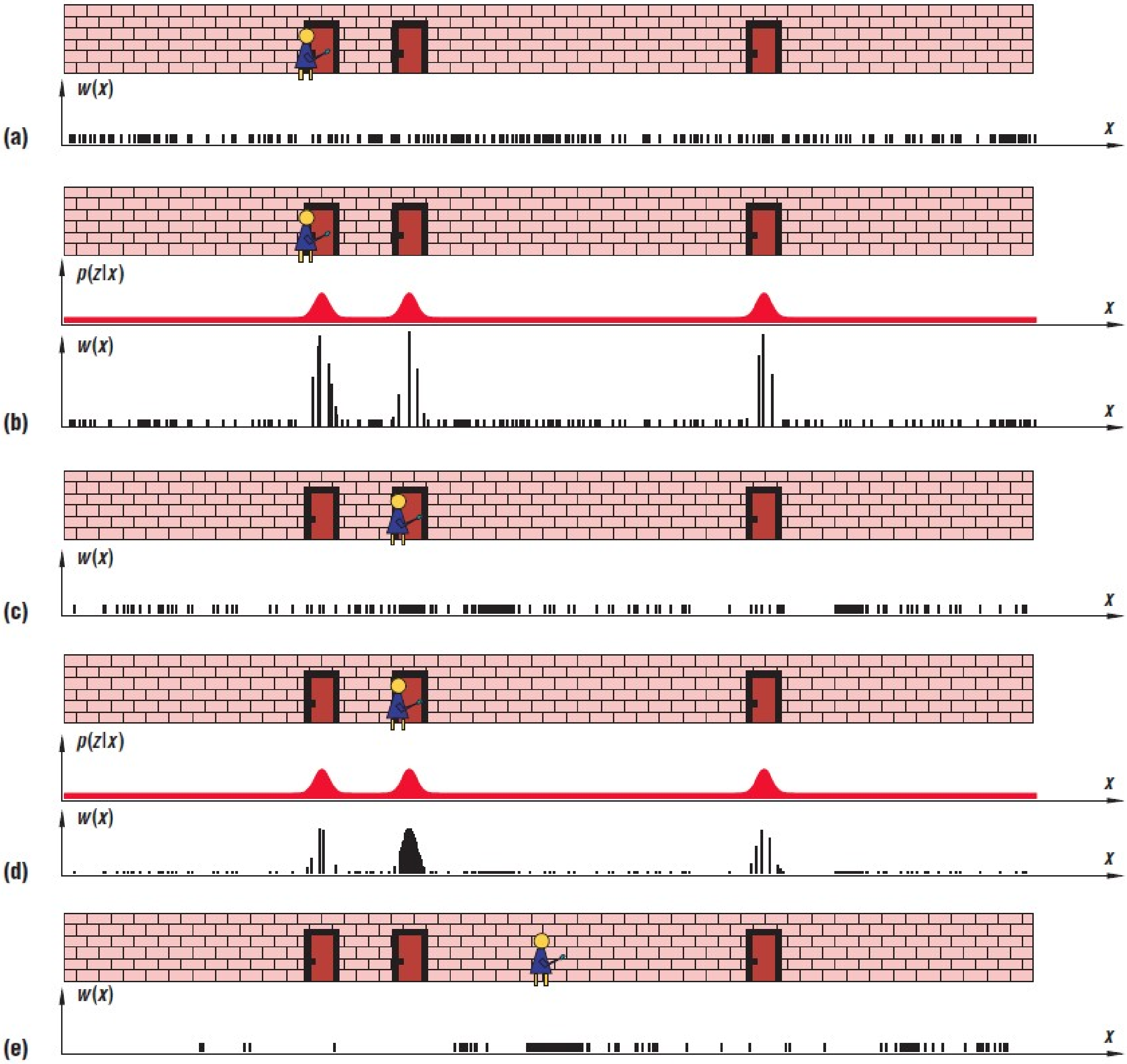

Figure 13: The black bars depict the particles representing the belief. a) A uniformly distributed sample set shows the person's initially unknown position. b) Sensor detects a door. The sample set is obtained from weighing the importance factors in proportion to the likelihood of measurement. c) An implementation of prediction step. The samples were drawn from the previous set with probability proportional to the importance factors (Re-sampling). d) Sensor detects a door. e) The sample set after another prediction step (Re-sampling).