Map Building with Mobile Robots in Dynamic Environments

Dirk Hähnel[†‡] Rudolph Triebel[‡] Wolfram Burgard[†‡] Sebastian Thrun[‡]

[†]University of Freiburg, Department of Computer Science, Germany
[‡]Carnegie Mellon University, School of Computer Science, PA, USA

Abstract

The problem of generating maps with mobile robots has received considerable attention over the past years. Most of the techniques developed so far have been designed for situations in which the environment is static during the mapping process. Dynamic objects, however, can lead to serious errors in the resulting maps such as spurious objects or misalignments due to localization errors. In this paper we consider the problem of creating maps with mobile robots in dynamic environments. We present a new approach that interleaves mapping and localization with a probabilistic technique to identify spurious measurements. In several experiments we demonstrate that our algorithm generates accurate 2d and 3d in different kinds of dynamic indoor and outdoor environments. We also use our algorithm to isolate the dynamic objects and to generate three-dimensional representation of them.

1 Introduction

Learning maps with mobile robots is one of the fundamental problems in mobile robotics. In the literature, the mobile robot mapping problem is often referred to as the *simultaneous localization and mapping problem (SLAM)* [5, 7, 13, 17, 14, 10, 19]. This is because mapping includes both, estimating the position of the robot relative to the map and generating a map using the sensory input and the estimates about the robot's pose.

Whereas most of todays mapping systems are able to deal with noise in the odometry and noise in the sensor data, they assume that the environment is static during mapping. However, if a person walks through the sensor range of the robot during mapping, the resulting map will contain evidence about an object at the corresponding location. Moreover, if the robot returns to this location and scans the area a second time, pose estimates will be less accurate, since the new measurement does not contain any features corresponding to the person. The reduced accuracy of the resulting maps may have a negative influence on the overall performance of the robot, since it can obstruct the execution of typical navigation tasks such as localization and path planning.

In this paper we present a new algorithm to mapping with mobile robots in dynamic environments. Our approach applies the popular Expectation-Maximization (EM) algorithm. In the expectation step we compute a probabilistic estimate about which measurements might correspond to static objects. In the maximization step we use these estimates to determine the position of the robot and the map. This process is iterated until no further improvement can be achieved.

We apply our approach to 2d and 3d data obtained with laser-range scanners. In practical experiments we demonstrate that our algorithm can reliably filter out dynamic aspects and yields accurate models of the environment. A further advantage of our algorithm is that the filtered data can be extracted from the rest of all measurements. This way, we can obtain accurate textured 3d models of dynamic objects.

This paper is organized as follows. After discussing related work in the following section, we will present our EM-based procedure to learn which measurements correspond to static aspects of the environment in Section 3. In Section 4 we will present several experiments illustrating that our approach can successfully learn 2d and 3d maps with range scanners in dynamic environments.

2 Related Work

For mobile robots, there exist several approaches to mapping in dynamic environments. The approaches mostly relevant to the work reported here are the methods developed by Wang et al. [20] and our previous work described in [11]. Wang et al. [20] use a heuristic and feature-based approach to identify dynamic objects in range scans. The corresponding measurements are then filtered out during 2d scan registration. In our recent work [11] we describe an approach to track persons in range scans and to remove the corresponding data during the registration and mapping process. Compared to these techniques, our algorithm presented in this paper does not rely on any pre-defined features. Rather, it considers every measurement individually and estimates a posterior about whether or not this data item has been generated by a dynamic object.

Additionally, there has been work on updating maps or improving localization in populated environments. For example, in the system described in [4] we update a given static map using the most recent sensory input to deal with people in the environment during path planning. Montemerlo et al. [15] present an approach to simultaneous localization and people tracking. Siegwart et al. [18] present a team of tour-guide

robots that operates in a populated exhibition. Their system uses line features for localization and has been reported to successfully filter range-measurements reflected by persons. Fox et al. [9] present a probabilistic technique to identify range measurements that do not correspond to the given model of the environment. These approaches, however, require a given and fixed map which is used for localization and for the extraction of the features corresponding to the people. Our technique, in contrast, does not require a given map. Rather it learns the map from scratch using the data acquired with the robot's sensors. Our algorithm repeatedly interleaves the process of estimating which beams are caused by dynamic objects with a mapping and localization algorithm. Thereby it iteratively improves its estimates and generates more accurate models of the environment.

From a more general perspective, the problem of estimating dynamic aspects in data can be regarded as an outlier detection problem, since the spurious measurements are data items that do not correspond to the static aspects of the environment which are to be estimated. The identification of outliers is an important subtask in various application areas such as data mining [12, 3, 16], correspondence establishment [6, 2], clustering [8], or statistics [1]. In all these fields, errors in the data reduce the accuracy of the resulting models and thus can lead to a decreased performance whenever the learned models are used for prediction or robot navigation, for example. The problem considered in this paper differs from these approaches in the fact that outliers cannot be detected solely based on their distance to the other data items. Rather, the measurements first have to be interpreted and transformed into a global representation (map) before individual measurements can be identified as outliers.

3 Learning Maps in Dynamic Environments

Our approach to discover measurements that correspond to dynamic objects is strictly statistical. We use the popular EM-algorithm to identify data items that cannot be explained by the rest of the data set. The input to our routine is a sequence of data items $z = \{z_1, \ldots, z_T\}$. The output is a model m obtained from these data items after incorporating the estimates about spurious measurements. In essence, our approach seeks to identify a model m that maximizes the likelihood of the data. Throughout this paper we assume that each measurement z_t consists of multiple data $z_{t,1}, \ldots, z_{t,N}$ as it is the case, for example, for laser-range scans. Throughout this paper we assume that the data $z_{t,n}$ are beams obtained with a laser-range scanner.

To accurately map a dynamic environment we need to know which measurements are caused by dynamic objects and therefore can safely be ignored in the alignment and map updating phase. To characterize spurious measurements in the data we introduce additional variables $c_{t,n}$ that tell us for each t and each n, whether the data item $z_{t,n}$ is caused by a static object or not. Each such variable $c_{t,n}$ is a binary variable, that is either 0 or 1. It is 1 if and only if the $z_{t,n}$ is caused by a static object. The vector of all these variables will be denoted by c.

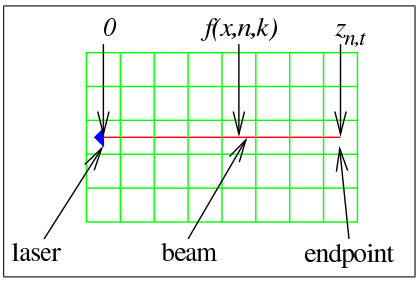

Figure 1: Beam covering $z_{t,n}$ cells of a map.

For the sake of simplicity, we give the derivation for beams that are parallel to the x-axis of the map. In this case, the length $z_{t,n}$ directly corresponds to the number of cells covered by this beam. We will later describe how to deal with beams that are not parallel to the x-axis. Let f be a function that returns for each position x_t of the robot, each beam number n, and each $k \leq z_{t,n}$ the index $f(x_t, n, k)$ of k-th field covered by that beam in the map (see Figure 1). To determine whether or not a beam is reflected by a dynamic object, we need to define the likelihood of a measurement given the current map m of the environment, the pose x of the robot, and the information about whether $z_{t,n}$ is reflected by a maximum range reading. Typically, maximum-range readings have to be treated differently, since those measurements generally are not reflected by any object. Throughout this paper we introduce indicator variables $\zeta_{t,n}$ which are 1 if and only if $z_{t,n}$ is a maximum range reading and 0, otherwise. The likelihood of a measurement $z_{t,n}$ given the value of $c_{t,n}$ and the map m can thus be computed as:

$$
\begin{aligned}
p(z_{t,n} \mid c_{t,n}, x_t, m) = & \left[\prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t,n,k)}) \right]^{\zeta_{t,n}} \\
& \cdot \left[[m_{f(x_t,n,z_{t,n})}]^{c_{t,n}} \cdot [1 - m_{f(x_t,n,z_{t,n})}]^{(1-c_{t,n})} \right. \\
& \left. \cdot \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t,n,k)}) \right]^{(1-\zeta_{t,n})} \qquad (1)
\end{aligned}
$$

The first term of this equation specifies the likelihood of the beam given it is a maximum range scan. In such a situation, we compute the likelihood as the product of the probabilities that the beam has covered the cells 0 to $z_{t,n-1}$. Please note, that the cell in which the beam ends does not provide any information since we do not know, whether there is an object or not given the beam is a maximum range reading. Thereby the probability that a beam covers a cell $k < z_{t,n}$ is equal to $1 - m_{f(x_t,n,k)}$. The second row of this equation specifies how to deal with the case that a cell that reflects a non-maximum range beam. If $z_{t,n}$ is not reflected by a dynamic object, i.e. $c_{t,n} = 1$, then the likelihood equals $m_{f(x_t,n,z_{t,n})}$. If, in contrast, $z_{t,n}$ is reflected by a dynamic object, the likelihood is $1 - m_{f(x_t,n,z_{t,n})}$. As well as for the maximum range measurements we have to

consider in both cases that the beam has covered $z_{t,n} - 1$ cells before reaching cell $f(x_t, n, z_{t,n})$.

Based on the definition of the observation likelihood we now will define the likelihood $p(z, c \mid x, m)$ of the data which we try to maximize in order to find the most likely map of the environment.

$$\begin{aligned} p(z, c \mid x, m) &= \prod_{t=1}^{T} p(z_t, c_t \mid x_t, m) && (2) \\ &= \prod_{t=1}^{T} p(z_t, \mid x_t, m) \cdot p(c_t \mid x_t, m) && (3) \\ &= \prod_{t=1}^{T} p(z_t, \mid x_t, m) \cdot p(c_t) && (4) \\ &= \prod_{t=1}^{T} \prod_{n=1}^{N} p(z_{t,n}, \mid c_{t,n}, x_t, m) \cdot p(c_t) && (5) \end{aligned}$$

We obtain Equation (3) from Equation (2) by assuming that the z_t and c_t are independent given x_t and m. We furthermore consider c_t as independent from the location x_t and the map m, which leads to Equation (4). Finally, Equation (5) is derived from Equation (4) under the usual assumption, that the neighboring beams of a single scan are independent given the map of the environment.

Maximizing $p(z, c \mid x, m)$ is equivalent to maximizing the corresponding log likelihood, which can be derived from Equation (5) and Equation (1) by straightforward mathematical transformations:

$$\begin{aligned} &\ln p(z, c \mid x, m) \\ &= \ln \prod_{t=1}^{T} \prod_{n=1}^{N} p(z_{t,n}, \mid c_{t,n}, x_t, m) \cdot p(c_t) \\ &= N \cdot \sum_{t=1}^{T} \ln p(c_t) + \sum_{t=1}^{T} \sum_{n=1}^{N} \ln p(z_{t,n}, \mid c_{t,n}, x_t, m) \\ &= N \cdot \sum_{t=1}^{T} \ln p(c_t) \\ &\quad + \sum_{t=1}^{T} \sum_{n=1}^{N} \Bigg[(1 - \zeta_{t,n}) \cdot \Big[c_{t,n} \cdot \ln m_{f(x_t,n,z_{t,n})} \\ &\qquad + (1 - c_{t,n}) \cdot \ln(1 - m_{f(x_t,n,z_{t,n})}) \Big] \\ &\qquad + \sum_{k=0}^{z_{t,n}-1} \ln(1 - m_{f(x_t,n,k)}) \Bigg] \end{aligned} \quad (6)$$

Since the correspondence variables c are not observable in the first place a common approach is to integrate over them, that is, to optimize the expected log likelihood $E_c[\ln p(c, z \mid x, m) \mid x, m, d]$ instead. Since the expectation is a linear operator, we can move it inside the expression. By exploiting the fact that the expectation of $c_{t,n}$ only depends on the corresponding measurement $z_{t,n}$ and the position x_t of the robot at that time. we can derive the following equation:

$$\begin{aligned} &E_c[\ln p(z, c \mid x, m) \mid z, x, m] = \\ &\quad \gamma + \sum_{t=1}^{T} \sum_{n=1}^{N} \Bigg[e_{t,n} \cdot (1 - \zeta_{t,n}) \cdot \ln m_{f(x_t,n,z_{t,n})} \\ &\quad + (1 - e_{t,n}) \cdot (1 - \zeta_{t,n}) \cdot \ln(1 - m_{f(x_t,n,z_{t,n})}) \\ &\quad + \sum_{k=0}^{z_{t,n}-1} \ln(1 - m_{f(x,n,k)}) \Bigg] \end{aligned} \quad (7)$$

For the sake of brevity, we use the term

$$e_{t,n} = E_c[c_{t,n} \mid z_{t,n}, x_t, m] \quad (8)$$

in this equation. The term

$$\gamma = N \cdot \sum_{t=1}^{T} E_c[\ln p(c_t) \mid z, x, m] \quad (9)$$

is computed from the prior $p(c_t)$ of the measurements which is independent of z, x, and m. Accordingly, γ can be regarded as a constant.

Unfortunately, optimizing Equation (7) is not an easy endeavor. A typical approach to maximize log likelihoods is the EM algorithm. In the particular problem considered here this amounts to generating a sequence of maps m of increasing likelihood. In the E-Step, we compute the expectations about the hidden variables c. In the M-step we then compute the most likely map m using the expectations computed in the E-Step. Both steps are described in detail in the remainder of this section.

In the E-step we compute the expectations $e_{t,n} = E_c[c_{t,n} \mid z_{t,n}, x_t, m]$ for each $c_{t,n}$ given the measurement $z_{t,n}$, the location x_t of the robot and the current map m. Exploiting the fact that $e_{t,n}$ equals $p(c_{t,n} \mid z_{t,n}, x_t, m)$ and considering the two cases that $z_{t,n}$ is a maximum range reading or not, we obtain:

$$e_{t,n} = \begin{cases} p(c_{t,n}) & \text{, if } \zeta_{t,n} = 1 \\ p(c_{t,n}) \epsilon_{t,n} & \text{, otherwise} \end{cases}$$

where

$$\epsilon_{t,n} = \frac{1}{p(c_{t,n}) + (1 - p(c_{t,n}))(\frac{1}{m_{f(x_t,n,z_{t,n})}} - 1)} \quad (10)$$

The first equation corresponds to the situation that $z_{t,n}$ is a maximum range reading. Then, $e_{t,n}$ corresponds to the prior probability $p(c_{t,n})$ that a measurement is reflected by a static object. Thus, a maximum range reading does not provide any evidence about whether or not the cell in the map in which the beam ends is covered by a dynamic object.

In the M-Step we want to determine the values for m and x that maximize Equation (7) after computing the expectations $e_{t,n}$

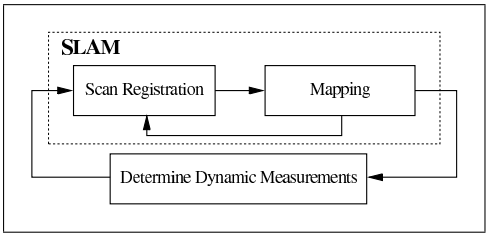

Figure 2: Iteration of SLAM and dynamic beam estimation.

about the hidden variables $c_{t,n}$ in the E-step. Unfortunately, maximizing this equation is also not trivial since it involves a solution to a high-dimensional state estimation problem. To deal with the enormous complexity of the problem, many researchers phrase it as an incremental maximum likelihood process [19, 10]. The key idea of incremental approaches is to calculate the desired sequence of poses and the corresponding maps by maximizing the marginal likelihood of the t-th pose and map relative to the $(t-1)$-th pose and map. In our algorithm, we additionally consider the estimations $e_{t,n}$ that measurement n at time t is caused by a static object of the environment:

$$\begin{aligned} \hat{x}_t &= \operatorname*{argmax}_{x_t} \{p(z_t \mid c_t, x_t, \hat{m}^{[t-1]}) \\ &\quad \cdot p(x_t \mid u_{t-1}, \hat{x}_{t-1})\} \end{aligned} \tag{11}$$

In this equation the term $p(z_t \mid c_t, x_t, \hat{m}^{[t-1]})$ is the likelihood of the measurement z_t given the pose $\hat{x}_t$ and the map $\hat{m}^{[t-1]}$ constructed so far. The term $p(x_t \mid u_{t-1}, \hat{x}_{t-1})$ represents the probability that the robot is at location x_t given the robot previously was at position $\hat{x}_{t-1}$ and has carried out (or measured) the motion u_{t-1}. The registration procedure is then carried out using the same algorithm as described in our previous work [11].

It remains to describe how the measurement z_t is then used to generate a new map $\hat{m}^{[t]}$ given the resulting pose $\hat{x}_t$ and the expectations $e_{t,n}$. Fortunately, once $x_1, \ldots, x_t$, have been computed, we can derive a closed-form solution for $m^{[t]}$. We want to determine the value of each field j of the map $m^{[t]}$ such that the overall likelihood of $m^{[t]}$ is maximized. To achieve this, we sum over individual fields $j \in [1, \ldots, J]$ of the map. Thereby we use an indicator function $I(y)$ which is 1, if y is true and 0, otherwise.

$$\begin{aligned} \hat{m}^{[t]} = \operatorname*{argmax}_m \Bigg(\sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \Bigg[& I(f(x_t, n, z_{t,n}) = j) \\ & \cdot (1 - \zeta_{t,n}) \cdot (e_{t,n} \ln m_j + (1 - e_{t,n}) \ln(1 - m_j)) \\ & + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \Bigg] \Bigg) \end{aligned} \tag{12}$$

Now suppose, we define

$$\tilde{I}(x, n, k, j) \quad := \quad I(f(x, n, k) = j)$$

and

$$\alpha_j \quad := \quad \sum_{t=1}^{T} \sum_{n=1}^{N} \tilde{I}(x_t, n, z_{t,n}, j) \cdot (1 - \zeta_{t,n}) \cdot e_{t,n}$$

$$\begin{aligned} \beta_j \quad := \quad \sum_{t=1}^{T} \sum_{n=1}^{N} \Bigg(& \tilde{I}(x_t, n, z_{t,n}, j) \cdot (1 - \zeta_{t,n}) \\ & \cdot (1 - e_{t,n}) + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \Bigg) \end{aligned}$$

The quantity α_j corresponds to the sum of the expectations $e_{t,n}$ that beam n of scan t is reflected by a static object of all beams that are not maximum-range beams and that end in cell j. The term β_j, on the other hand, is the sum of two terms. The first term is the sum of the expectations $1 - e_{t,n}$ that beam n of scan t is reflected by a dynamic object of all beams that are not maximum-range beams and that end in cell j. The second value of the sum simply is the number of times a beam covers j but does not end in j. Please note that this value is independent from whether or not the corresponding beam is reflected by a dynamic object or not. Please furthermore note that in a static world with $e_{t,n} = 1$ for all t and n the term α_t corresponds to the number of times a beam that does not have the maximum length ends in j. In contrast to that, β_j is the number of times a beam covers a cell.

Using the definitions of α_j and β_j, Equation (12) turns into

$$m^{[t]} = \operatorname*{argmax}_m \left(\sum_{j=1}^{J} \alpha_j \ln m_j + \beta_j \ln(1 - m_j) \right) \tag{13}$$

Since all m_j are independent, we maximize the overall sum by maximizing each m_j. A necessary condition to ensure that m_j is a maximum is that the first derivative equals zero:

$$\frac{\partial m}{\partial m_j} = \frac{\alpha_j}{m_j} - \frac{\beta_j}{1 - m_j} = 0 \tag{14}$$

By straightforward mathematical transformationswe obtain

$$m_j \quad = \quad \frac{\alpha_j}{\alpha_j + \beta_j}. \tag{15}$$

Please note that, given the sensor model specified in Equation (1), this closed-form solution for the most likely map m for given positions x and static environments corresponds to the naive counting technique in which one counts for each cell how often a beam has ended in that cell and how often a beam has covered it without ending in it.

The overall approach can be summarized as follows (see also Figure 2). We start with an initial map $\hat{m}$ obtained by the incremental mapping approach. Thereby the expectations $e_{t,n}$ are initialized with the prior probability $p(c_{t,n})$ that a measurement is caused by a static object. Given the resulting map $\hat{m}$ and the corresponding positions $\hat{x}$, we compute new expectations $e_{t,n}$ for each beam according to Equation (8). These expectations are then used to compute a new map. The overall process is iterated until no improvement of the overall likelihood (Equation (6)) can be achieved or a certain number of iterations has been exceeded.

Finally, we would like to discuss how to deal with beams that are not parallel to the x-axis. In this case we no longer can

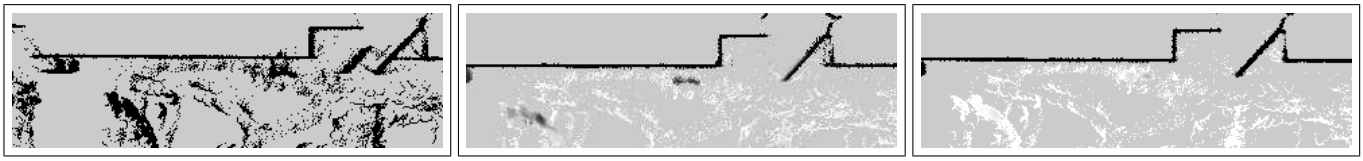

Figure 5: Evolution of the map during EM. The images corresponds to iteration 1, 2, and 6.

Figure 3: Robot Sam mapping the populated exhibition hall of the Byzantine Museum in Athens (left). In the resulting map (right), the measurements labeled as dynamic are shown in grey/orange.

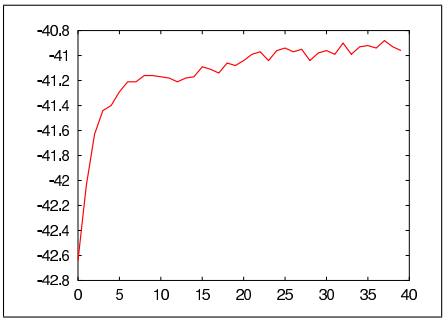

Figure 4: Evolution of the log likelihood (Equation (6)) during the individual iterations.

compute the likelihood that a beam covers a cell j of m as $(1 - m_j)$. Otherwise, transversal beams covering more cells would accumulate a lower likelihood. The solution to this is to weigh the beams according to the length by which they cover a cell. Suppose B is the set of cells in m covered by a beam. Furthermore suppose l_j is the length by which the beam covers field $j \in B$. Then, the likelihood of a covering all cells in B is computed as $\prod_{j \in B} (1 - m_j)^{l_j}$.

4 Experiments

The approach described above has been implemented and tested on different robotic platforms, in different environments and with 2d and 3d data. In all experiments, we figured out, that the system is robust even in highly dynamic environments. In one experiment carried out with a fast moving car, the system was able to accurately map the environment even if no odometry data was given.

4.1 Filtering People

The first experiments were carried out using the Pioneer 2 robot Sam in the empty exhibition hall of the Byzantine Museum in Athens, Greece. The size of this environment is 30m x 45m. The robot traveled continously 57m with an avg. speed of 0.37m/s and a max. speed of 0.96m/s. Figure 3 (left) shows the robot during the mapping process. There were 15 people walking with a typical speed through the environment while the robot was mapping it. Partially they stopped and continued moving. The most likely map resulting from the application of our approach is shown in athe right image of Figure 3. The beams labeled as dynamic are drawn grey/orange in this figure. As can be seen, our approach can reliably identify dynamic aspects and is able to learn maps that include the static aspects only. At this point we would also like to mention that the resulting map contains seriously less dynamic objects than the map obtained with our previous approach presented in [11].

Figure 4 plots the evolution of $E_c[\ln p(c, z \mid x, m) \mid x, m, d]$ over the different iterations of our algorithm. It illustrates that our algorithm in fact maximizes the overall log likelihood. Please note, that this curve generally is not monotonic, because of the incremental maximum-likelihood solution to the SLAM problem. Slight variations in the pose can have negative effects in future steps, so that the map likelihood can decrease. However, we never observed significant decrease of the log likelihood.

4.2 Improved Localization Accuracy

Besides the fact that the resulting maps contain less spurious objects, our approach also increases the localization accuracy. If dynamic objects are not handled appropriately during localization, matching errors become more likely. Figure 6 shows a typical map we obtained when mapping a densely populated environment. In this case we mapped a part of the Wean Hall Corridor at Carnegie Mellon University during peak office hours when many persons were around. Some of them were trying to block the robot, so that the robot had to make detours around them. Therefore the robot traveled 74m with an avg. speed of 0.15m/s (0.35m/s maximum). Despite the fact, that the huge amount of spurious objects make the map virtually useless for navigation tasks, the map also shows serious errors in the alignment. Some of the errors are indicated by arrows in the corresponding figure.

Figure 7 shows the map generated by our algorithm. As the figure illustrates, the spurious measurements (indicated by grey/orange dots) have been filtered out completely. Additionally, the alignment of the scans is more accurate.

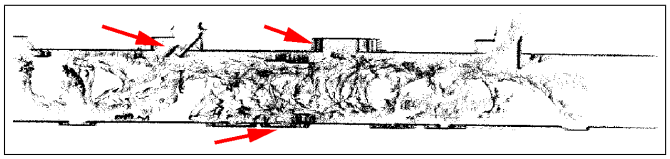

Figure 6: Map obtained in a populated corridor of the Wean Hall at Carnegie Mellon University using the raw input data.

Figure 7: Map generated by our algorithm.

Figure 5 depicts the evolution of a part of the map in the different rounds of the EM. It shows how the beams corresponding to dynamic objects slowly fade out and how the improved estimates about these beams improve the localization accuracy.

4.3 Generating Large-Scale Outdoor Maps

To evaluate the capability of our technique to deal with arbitrary features, we mounted a laser-range scanner on a car and drove approximately 1km through Pittsburgh, PA, USA (Corner between Craig Street and Forbes Avenue). The maximum speed of the car was 35 MPH in this experiment. We then applied our approach to the recorded data. The map generated by our algorithm is shown in Figure 8. Whereas the black dots correspond to the static objects in the scene, the white dots are those which are filtered out using our approach. Again, most of the dynamics of the scene could be removed. Only a few cars could not be identified as dynamic objects. This is mainly because we quickly passed cars waiting for turns and because we drove along the path only once. Please also note, that due to the lack of a GPS, the map had to be computed without any odometry information.

4.4 Generating Textured 3D Maps

To demonstrate that our approach is not limited to 2d range data, we carried out several experiments with the mobile robot Robin (see Figure 9) which is equipped with a laser-scanner

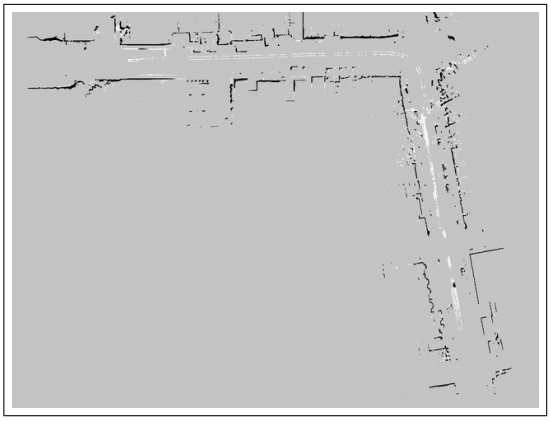

Figure 8: Map of an outdoor scene after filtering dynamic objects.

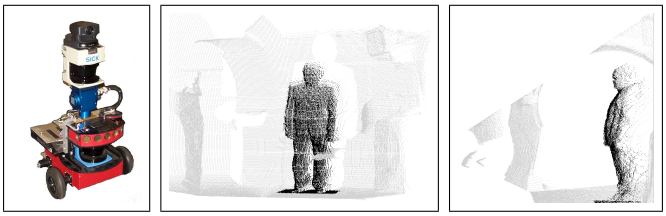

Figure 9: The mobile robot Robin used to generate textured 3d models (left). Beams reflected by a person are isolated from the rest of the data. This is achieved by computing a bounding box around those beams perceived with the horizontal scanner that are identified as corresponding to dynamic objects (center and right).

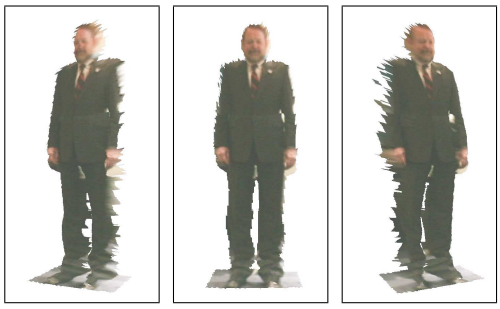

Figure 10: Textured 3d model of a person identified as a dynamic object.

mounted on an AMTEC pan/tilt unit. On top of this scanner we installed a camera which allows us to obtain textured 3d maps of an environment. Additionally, this robot contains a horizontally scanning laser range finder which we used in our experiments to determine dynamic objects. To label the beams in the 3d data as dynamic we use a bounding box around the dynamic 2d points. To filter dynamic objects in the textures recorded with Robin's cameras we choose for every polygon that image which has the highest likelihood of containing static aspects only. The left image of Figure 11 shows one particular view of a model obtained without filtering of dynamic objects. The arrow indicates a polygon whose texture contains fractions of an image of a person which walked through the scene while the robot was scanning it. After applying our approach the corresponding beams and parts of the pictures were filtered out. The resulting model shown in the right image of Figure 11 therefore only contains textures showing static objects.

4.5 Extracting Textured 3d Objects

Additionally to filtering dynamic objects and learning static aspects of environments our algorithm can also be used to separate dynamic objects from the environment. The key idea is to extract all measurements from the 3d data that lie within a bounding box around the beams whose probability that they are reflected by dynamic objects exceeds 0.7. Figure 9 shows two views of a typical 3d data sets obtained with this approach. Whereas the data points belonging to a dynamic object are shown in black, the rest of the data is depicted in grey. Again we used the camera to map textures on the 3d data that were

Figure 11: Textured 3d models obtained using Robin. The upper image shows the result without filtering. The lower image shows the resulting model obtained with our algorithm.

identified as belonging to a dynamic object. Figure 10 depicts three views of the resulting model. As can be seen from the figure, our approach can accurately extract realistic looking textured 3d models of dynamic objects.

5 Conclusions

In this paper we presented a probabilistic approach to mapping in dynamic environments. Our approach uses the EM algorithm to interleave the identification of measurements that correspond to dynamic objects with a mapping and localization algorithm. This way it incrementally improves its estimate about spurious measurements and the quality of the map. The finally obtained maps contain less spurious objects and also are more accurate because of the improved range registration.

Our technique has been implemented and tested on different platforms. In several experiments carried out in indoor and outdoor environments we demonstrated that our approach yields accurate maps even if used on a fast moving vehicle without odometry information. We also presented an application to learn textured 3d models of dynamic environments. Finally, we applied our algorithm to extract dynamic objects from 3d data. The results illustrate that our approach can reliably estimate which beams correspond to dynamic objects.

Acknowledgements

This work has partly been supported by the EC under contract number IST-2000-29456 and by the German Science Foundation (DFG) under contract number SFB/TR8-03. It has also been sponsored by DARPA's MARS, CoABS, and MICA Programme (contract numbers N66001-01-C-6018,NBCH1020014, F30602-98-2-0137, and F30602-01-C-0219) and by the NSF under grant numbers IIS-9876136 and IIS-9877033.

References

[1] V. Barnett and T. Lewis. *Outliers in Statistical Data*. Wiley, New York, 1994.

[2] P. Besl and N. McKay. A method for registration of 3d shapes. *Trans. Patt. Anal. Mach. Intell. 14(2)*, pages 239–256, 1992.

[3] C. E. Brodley and M. A. Friedl. Identifying and eliminating mislabeled training instances. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1996.

[4] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000.

[5] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós. The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–953, 1999.

[6] I.J. Cox and S.L. Hingorani. An efficient implementation of reids multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on PAMI*, 18(2):138–150, February 1996.

[7] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *ICRA'2000 Workshop on Mobile Robot Navigation and Mapping*, 2000.

[8] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.

[9] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)*, 11:391–427, 1999.

[10] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.

[11] D. Hähnel, D. Schulz, and W. Burgard. Mapping with mobile robots in populated environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[12] George H. John. Robust decision trees: Removing outliers from databases. In *First International Conference on Knowledge Discovery and Data Mining*, pages 174–179, 1995".

[13] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Proc. of the Ninth Int. Symp. on Robotics Research (ISRR)*, 1999.

[14] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

[15] M. Montemerlo and S. Thrun. Conditional particle filters for simultaneous mobile robot localization and people-tracking (slap). In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.

[16] S. Ramaswamy, R. Rastogi, and S. Kyuseok. Efficient algorithms for mining outliers from large data sets. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2000.

[17] H. Shatkay. *Learning Models for Robot Navigation*. PhD thesis, Computer Science Department, Brown University, Providence, RI, 1998.

[18] http://robotics.epfl.ch/, 2002.

[19] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

[20] C.-C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.