

# MOBILE ROBOT MAPPING IN POPULATED ENVIRONMENTS

Dirk Hähnel<sup>1</sup>, Dirk Schulz<sup>2</sup>, and Wolfram Burgard<sup>1</sup>

<sup>1</sup>University of Freiburg, Department of Computer Science, Germany

<sup>2</sup>University of Bonn, Department of Computer Science, Germany

**Abstract** — The problem of learning maps with mobile robots has received considerable attention over the past years. Most of the approaches, however, assume that the environment is static during the data-acquisition phase. In this paper we consider the problem of creating maps with mobile robots in populated environments. Our approach uses a probabilistic method to track multiple people and to incorporate the estimates of the tracking technique into the mapping process. The resulting maps are more accurate since the number of spurious objects is reduced and since the robustness of range registration is improved. Our approach has been implemented and tested on real robots in indoor and outdoor scenarios. We present several experiments illustrating the capabilities of our approach to generate accurate 2d and 3d maps.

**Keywords:** Map building, mobile robot navigation, people tracking, scan matching, dynamic environments

## 1 Introduction

Learning maps with mobile robots has received considerable attention over the last two decades. This is because maps often are inherently necessary for mobile robots to perform their tasks. When mapping an environment, a mobile robot generally has to cope with different kinds of noise: noise in the odometry and noise in the sensor data. Therefore, the map learning problem is a chicken-and-egg problem. If the pose (we use the term *pose* to refer to a robot's  $x$ - $y$  location and its heading direction  $\theta$ ) of the robot is always known during mapping, building maps is relatively easy. On the other hand, if a map is available, determining the robot's poses can be done efficiently. In the literature, the mobile robot mapping problem is often referred to as the *simultaneous localization and mapping problem (SLAM)* [7, 9, 21].

Whereas most of today's mapping systems are able to deal with noise in the odometry and noise in the sensor data, they usually 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. If the robot later 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. Thus, people walking through the scene result in spurious objects in the map and simultaneously make the localization problem harder. Moreover, 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 article, which is an extended version of our work published in [15], we present a new algorithm to mapping with mobile robots in populated environments. Our approach uses sample-based joint probabilistic data association filters for tracking people in the vicinity of the robot. The belief about the potential positions of persons is used to improve the registration process and to remove spurious objects from the map being learned. We apply our approach to 2d and 3d data obtained with laser-range scanners. In practical experiments, we demonstrate that our algorithm yields accurate models of the environment by filtering out beams reflected by persons.

This paper is organized as follows. After discussing related work in the following section, we will briefly describe our approach to tracking multiple people in range scans in Section 3. In Section 4 we will present our mapping technique and how the results of the people tracking approach are integrated into the mapping process. Finally, Section 5 contains several experiments illustrating the advantages of our approach to learning accurate 2d and 3d maps with range scanners.

## 2 Related Work

Approaches to concurrent mapping and localization can roughly be classified according to the kind of sensor data processed and the matching algorithms used. For example, the approaches described in [27, 7, 9, 21] extract

landmarks out of the data and match these landmarks to localize the robot in the map being learned. The other set of approaches such as [22, 13, 30] use raw sensor data and perform a dense matching of the scans. Although all approaches possess the ability to cope with a certain amount of noise in the sensor data, they assume that the environment is almost static during the mapping process. Especially in populated environments, additional noise is introduced to the sensor data which increases the risk of localization errors. Additionally, people in the vicinity of the robot appear as objects in the resulting maps and therefore make the maps not usable for path planning etc. Recently Wang and Thorpe [33] presented a heuristic and feature-based approach to identify dynamic objects in range scans. The corresponding measurements are then filtered out during 2d scan registration. Our approach instead uses a tracking technique and therefore is able to predict the positions of the persons even in situations in which the corresponding features are temporarily missing.

Additionally, there has been work on updating maps or improving localization in populated environments. For example, Burgard et al. [6] update a given static map using the most recent sensory input to deal with people in the environment during path planning. Montemerlo et al. [23] present an approach to simultaneous localization and people tracking. Arras et al. [1] 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. [11] 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.

Furthermore, there has been developed a variety of techniques for tracking persons, for predicting future poses of persons, or for adopting the behavior of the robot according to the information obtained about the persons in its vicinity [29, 19, 34, 20, 5, 17, 18, 16, 32, 4, 2, 28]. All these approaches, however, do not filter the measurements corresponding to persons in order to improve the model of the environment.

In this paper we present a probabilistic approach to filtering people out of sensor data and to incorporate the results of the filtering into the mapping process. Our approach has several desirable properties. First, by incorporating the results of the people tracker the alignment of the scans becomes more robust. Additionally, the resulting maps are more accurate, since measurements corrupted by people walking by are filtered out. Empirical results, described in this paper, illustrate that our approach succeeds in learning accurate large-scale 2d and 3d maps of populated environments with range scanners even if several persons are in the vicinity of the robot.

### 3 Tracking People in Range Scans

To detect people and track people in the vicinity of the robot, our system applies a sample-based variant of Probabilistic Data Association Filters (JPDAFs) [8]. This approach is described in the following section. Additionally, we will describe how we adapt the number of persons being tracked and how to implement this technique using data gathered with the laser-range finders of a moving mobile robot.

#### 3.1 Sample-based Joint Probabilistic Data Association Filters (SJPDAs)

Suppose there are  $K$  persons and let  $\mathbf{X}^t = \{\mathbf{x}_1^t, \dots, \mathbf{x}_K^t\}$  be the states of these persons at time  $t$ . Note that each  $\mathbf{x}_i^t$  is a random variable ranging over the state space of a single person. Furthermore, let  $\mathbf{Z}(t) = \{\mathbf{z}_1(t), \dots, \mathbf{z}_{m_t}(t)\}$  denote a feature set observed at time  $t$ , where  $\mathbf{z}_j(t)$  is one feature of such a set.  $\mathbf{Z}^t$  is the sequence of all feature sets up to time  $t$ . The key question when tracking multiple persons is how to assign the observed features to the individual objects.

In the JPDAF framework, a joint association event  $\theta$  is a set of pairs  $(j, i) \in \{0, \dots, m_t\} \times \{1, \dots, K\}$ . Each  $\theta$  uniquely determines which feature is assigned to which object. Please note, that in the JPDAF framework, the feature  $\mathbf{z}_0(t)$  is used to model situations in which an object has not been detected, i.e. no feature has been found for object  $i$ . Let  $\Theta_{ji}$  denote the set of all valid joint association events which assign feature  $j$  to the object  $i$ . At time  $t$ , the JPDAF considers the posterior probability that feature  $j$  is caused by object  $i$ :

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} P(\theta | \mathbf{Z}^t). \quad (1)$$

According to [26], we can compute the  $\beta_{ji}$  as

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} \alpha \gamma^{(m_t - |\theta|)} \prod_{(j,i) \in \theta} p(\mathbf{z}_j(t) | \mathbf{x}_i^t). \quad (2)$$

It remains to describe, how the beliefs  $p(\mathbf{x}_i^t)$  about the states of the individual objects are represented and updated. In our approach [26], we use sample-based representations of the individual beliefs. The key idea underlying all particle filters is to represent the density  $p(\mathbf{x}_i^t | \mathbf{Z}^t)$  by a set  $\mathbf{S}_i^t$  of  $N$  weighted, random samples or *particles*. A sample set constitutes a discrete approximation of a probability distribution. Each sample is a tuple  $(x_{i,n}^t, w_{i,n}^t)$  consisting of state  $x_{i,n}^t$  and an importance factor  $w_{i,n}^t$ . The *prediction* step is realized by drawing samples from the set computed in the previous iteration and by updating their state according to the prediction model  $p(\mathbf{x}_i^t | \mathbf{x}_i^{t-1}, \delta t)$ . In the *correction* step, a feature set  $\mathbf{Z}(t)$  is integrated into the samples obtained in the prediction step. Thereby we consider the assignment probabilities  $\beta_{ji}$ . In the sample-based variant, these quantities are obtained by integrating over all samples:

$$p(\mathbf{z}_j(t) | \mathbf{x}_i^t) = \frac{1}{N} \sum_{n=1}^N p(\mathbf{z}_j(t) | x_{i,n}^t). \quad (3)$$

Given the assignment probabilities we now can compute the weights of the samples

$$w_{i,n}^t = \alpha \sum_{j=0}^{m_t} \beta_{ji} p(\mathbf{z}_j(t) | x_{i,n}^t), \quad (4)$$

where  $\alpha$  is a normalizer ensuring that the weights sum up to one over all samples. Finally, we obtain  $N$  new samples from the current samples by bootstrap resampling. For this purpose we select every sample  $x_{i,n}^t$  with probability  $w_{i,n}^t$ .

### 3.2 Dealing with Variable Numbers of Persons

To adapt our algorithm to the number of people in the perceptual field of the robot we apply a Bayesian filtering process and maintain a density  $P(N^k | \mathbf{M}^k)$  over the number of objects  $N^k$  at time  $k$ , where  $\mathbf{M}^k = m_0, \dots, m_k$  is the sequence of the numbers of features observed so far:

$$P(N^k | \mathbf{M}^k) = \alpha \cdot P(m_k | N^k) \cdot \sum_n P(N^k | N^{k-1} = n) \cdot P(N^{k-1} = n | \mathbf{M}^{k-1}) \quad (5)$$

To implement this equation all we need to know are the quantities  $P(m_k | N^k)$  and  $P(N^k | N^{k-1})$ . The term  $P(m_k | N^k)$  represents the probability of observing  $m_k$  features, if  $N^k$  objects are in the perceptual field of the sensor. According to our previous work [26], this quantity is learned from data. To adapt the number of objects in the SJPDFAF, our system uses the maximum likelihood estimate of  $P(N^k | \mathbf{M}^k)$ . If the number of particle filters in the SJPDFAF is smaller than the new estimate for  $N^k$ , new filters need to be initialized. Since we do not know which of the features originate from new objects, we initialize the new filters using a uniform distribution. We then rely on the SJPDFAF to disambiguate this distribution during subsequent filter updates.

If the most recent estimate for  $N^k$  is smaller than the current number of particle filters, the corresponding number of filters needs to be decreased. This, however, requires that we know which filter does not track an object any longer. To estimate the tracking performance of a sample set, we accumulate a discounted average  $\hat{W}_i^k$  of the sum of sample weights  $W_i^k$  before the normalization step:

$$\hat{W}_i^k = (1 - \delta) \hat{W}_i^{k-1} + \delta W_i^k. \quad (6)$$

Since the sum of sample weights decreases significantly, whenever a filter is not tracking any feature contained in the measurement, we use this value as an indicator that the corresponding object has left the perceptual field of the robot. Whenever we have to remove a filter, we choose that one with the smallest discounted average  $\hat{W}_i^k$ .

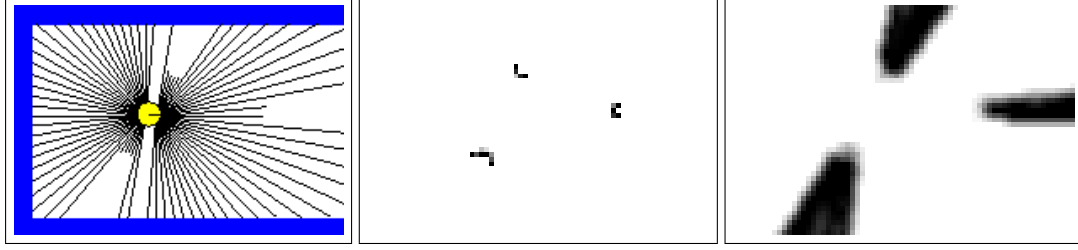


Figure 1. Typical laser range finder scan. Two of the local minima are caused by people walking by the robot (left image). Features extracted from the scan, the grey-level represents the probability that a person’s legs are at the position (center). Occlusion grid, the grey-level represents the probability that the position is occluded (right image).

### 3.3 Laser-based Implementation

In our system we apply the SJPDFAF to estimate the trajectories of persons in range scans. Since the laser range scanners mounted on our platforms are at a height of approx. 40 cm, the beams are reflected by the legs of the people which typically appear as local minima in the scans. These local minima are used as the features for the SJPDFAF (see left and center image of Figure 1). Unfortunately, there are other objects which produce patterns similar to people. To distinguish these static objects from moving people our system additionally considers the differences between occupancy probability grids built from consecutive scans. This whole process is illustrated in Figure 2. Please note, that we also perform a scan-matching to align each pair consecutive scans. Therefore, the static aspects of the environment can be identified and filtered out accurately.

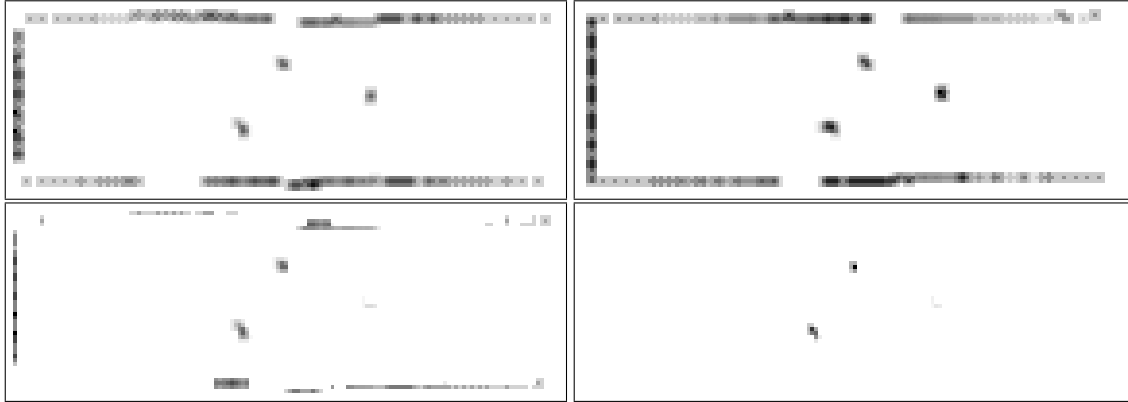


Figure 2. From left to right, top-down: the occupancy map for the current scan, the occupancy map for the previous scan, the resulting difference map, and the fusion of the difference map with the feature maps for the scan depicted in Figure 1.

Finally, we have to deal with possible occlusions. We therefore compute a so-called “occlusion map” containing for each position in the vicinity of the robot the probability that the corresponding position is not visible given the current range scan. See right part of Figure 1. The information about occluded areas is used to avoid that the SJPDFAF loses track of a person whenever it is temporarily occluded.

Figure 3 shows a typical situation, in which a robot equipped with two laser-range scanners is tracking up to four persons in its vicinity. As can be seen from the figure, our approach is robust against occlusions and can quickly adapt to changing situations in which additional persons enter the scene. For example, in the lower left image the upper right person is not visible in the range scan, since it is occluded by the person that is close to the robot. The knowledge that the samples lie in an occluded area prevents the robot from deleting the corresponding sample set. Instead, the samples only spread out, which represents the growing uncertainty of the robot about the position of the person.

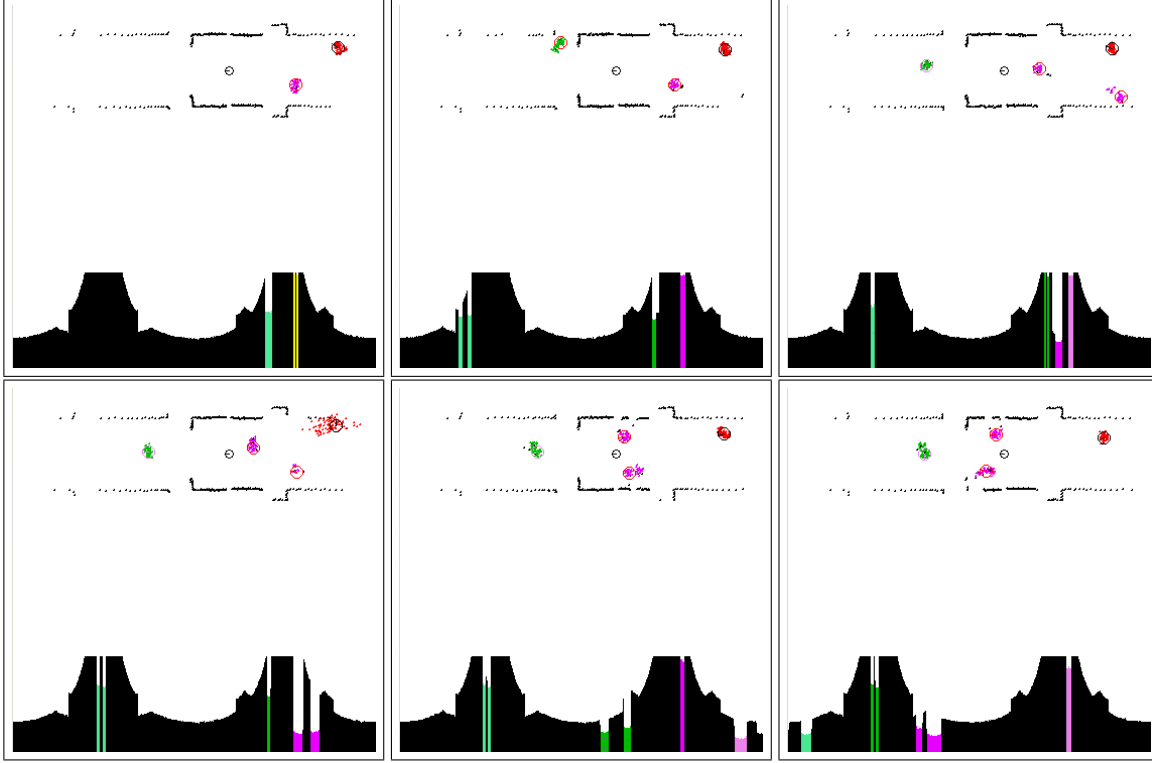


Figure 3. Tracking people using laser range-finder data. Upper part shows the raw sensor data with the colored sample sets. Lower part shows the range values with the local minima marked with the color of the corresponding sample set.

## 4 Computing Consistent Maps

Our current system is able to learn 2d and 3d maps using range scans recorded with a mobile robot. In both cases, the approach is incremental. Mathematically, we calculate a sequence of poses  $\hat{l}_1, \hat{l}_2, \dots$  and corresponding maps by maximizing the marginal likelihood of the  $t$ -th pose and map relative to the  $(t-1)$ -th pose and map:

$$\hat{l}_t = \underset{l_t}{\operatorname{argmax}} \{p(s_t | l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1})) \cdot p(l_t | u_{t-1}, \hat{l}_{t-1})\} \quad (7)$$

In this equation the term  $p(s_t | l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1}))$  is the probability of the most recent measurement  $s_t$  given the pose  $l_t$  and the map  $\hat{m}(\hat{l}^{t-1}, s^{t-1})$  constructed so far. The term  $p(l_t | u_{t-1}, \hat{l}_{t-1})$  represents the probability that the robot is at location  $l_t$  given the robot was previously at position  $\hat{l}_{t-1}$  and has carried out (or measured) the motion  $u_{t-1}$ . The resulting pose  $\hat{l}_t$  is then used to generate a new map  $\hat{m}$  via the standard incremental map-updating function presented in [24]:

$$\hat{m}(\hat{l}^t, s^t) = \underset{m}{\operatorname{argmax}} p(m | \hat{l}^t, s^t) \quad (8)$$

The overall approach can be summarized as follows: At any point  $t-1$  in time the robot is given an estimate of its pose  $\hat{l}_{t-1}$  and a map  $\hat{m}(\hat{l}^{t-1}, s^{t-1})$ . After the robot moved further on and after taking a new measurement  $s_t$ , the robot determines the most likely new pose  $\hat{l}_t$ . It does this by trading off the consistency of the measurement with the map (first term on the right-hand side in (7)) and the consistency of the new pose with the control action and the previous pose (second term on the right-hand side in (7)). The map is then extended by the new measurement  $s_t$ , using the pose  $\hat{l}_t$  as the pose at which this measurement was taken.

It remains to describe how we actually maximize Equation (7). Our system applies two different approaches depending on whether the underlying scans are 2d or 3d scans.

## 4.1 Two-dimensional Scan Alignment

Our algorithm used for 2d scan matching is an extension of the approach presented in [31]. To align a scan relative to the map constructed so far, we compute an occupancy grid map  $\hat{m}(\hat{l}^{t-1}, s^{t-1})$  [10, 24, 31] out of the sensor scans obtained so far. Additionally we integrate over small Gaussian errors in the robot pose when computing the maps. This avoids that many cells remain unknown especially if  $t$  is small. Additionally, it increases the smoothness of the map and corresponding likelihood function to be optimized and thus facilitates the range registration. As an example, consider the left image of Figure 4 which shows a typical map constructed out of 100 scans. The darker a location, the more likely it is that the corresponding place in the environment is covered by an obstacle. Please note that the map appears slightly blurred according to the integration over small pose errors. To maximize the likelihood of a scan with respect to this map, we apply a hill climbing strategy. A typical scan is shown in the center of Figure 4. The optimal alignment of this scan with respect to the map is shown in the right image of Figure 4. As can be seen from the figure, the alignment is quite accurate.



Figure 4. Two-dimensional scan alignment: Map created so far (left image), measurement  $s_t$  obtained at time  $t$  (center image) and resulting alignment (right image).

## 4.2 Aligning Three-dimensional Range Scans

Unfortunately, a three-dimensional variant of the maps used for the 2d scan alignment would consume too much memory in the case of three dimensions. Therefore this approach is not applicable to 3d scan alignment. Instead, we represent the 3d maps as triangle meshes constructed from the individual scans. We create a triangle for three neighboring scan points, if the maximum length of an edge does not exceed a certain threshold which depends on the length of the beams.

To compute the most likely position of a new 3d scan with respect to the current 3d model, we apply an approximative physical model of the range scanning process [11, 14]. Obviously, an ideal sensor would always measure the correct distance to the closest obstacle in the sensing direction. However, sensors and models generated out of range scanners are noisy. Therefore, our systems incorporates measurement noise and random noise to deal with errors typically found in 3d range scans. First, we generally have normally distributed measurement errors around the distance “expected” according to the current position of the scanner and the given model of the environment. Additionally, we observe randomly distributed measurements because of errors in the model and because of deviations in the angles between corresponding beams in consecutive scans. Therefore, our model consists of a mixture of a Gaussian with a uniform distribution. The mode of the Gaussian corresponds to the expected distance given the current map. Additionally, we use a uniform distribution to deal with maximum range readings. To save computation time, we approximate the resulting distribution by a mixture of triangular distributions.

Whereas this approach saves memory, it requires more computation time than the technique for 2d scan alignment. However, in practical experiments we found out that this technique has two major advantages over the Iterative Closest Point (ICP) algorithm [3, 12] which are crucial in the context of 3d mapping. First, it exploits the fact that each laser beam is a ray that does not go through surfaces and therefore does not require special heuristics for dealing with occlusions. Second, our approach also exploits the information provided by maximum range beams if they go through surfaces in the map since they reduce the likelihood of an alignment

To compute the likelihood of a beam  $b$  given the current map  $\hat{m}(\hat{l}^{t-1}, s^{t-1})$ , we first determine the expected distance  $e(b, \hat{m}(\hat{l}^{t-1}, s^{t-1}))$  to the closest obstacle in the measurement direction. This is efficiently carried out using ray-tracing techniques based on a spatial tiling and indexing [25] of the current map.

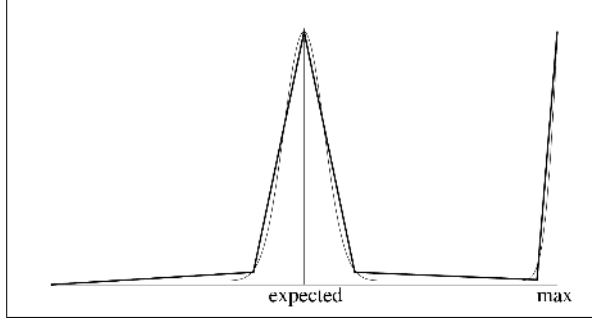


Figure 5. The probabilistic measurement model given as a mixture of a Gaussian and a uniform distribution and its approximation by piecewise linear functions.

Then we compute the likelihood of the measured distance given the expected distance, i.e. we determine the quantity  $p(b | e(b, \hat{m}(\hat{l}^{t-1}, s^{t-1})))$  using the mixture computed for  $e(b, \hat{m}(\hat{l}^{t-1}, s^{t-1}))$ . To speed up computation, we approximate this density by piecewise linear functions (see also Figure 5). Assuming that the beams contained in  $s_t$  are independent, we compute the likelihood of the whole scan as

$$p(s_t | l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1})) = \prod_{b \in s_t} p(b | e(b, \hat{m}(\hat{l}^{t-1}, s^{t-1}))). \quad (9)$$

To maximize Equation 7 we again apply a hill climbing technique.

### 4.3 Integrating People Tracking Results into the Map Building Process

The goal of integrating the results of the people tracker into a mapping process can be divided in two subjects:

1. to improve the alignment between the scans and
2. to filter out corrupted measurements originating from people walking in the vicinity of the robot.

To consider the estimated states of the persons during the scan alignment, we need to know the probability  $P(\text{hit}_{x,y} | \mathbf{X}^t)$  that a beam ending at position  $\langle x, y \rangle$  is reflected by a person. In our current implementation, we consider the individual persons independently:

$$P(\text{hit}_{x,y} | \mathbf{X}^t) = 1 - \prod_{i=1}^K (1 - P(\text{hit}_{x,y} | \mathbf{x}_i^t)). \quad (10)$$

In this equation  $P(\text{hit}_{x,y} | \mathbf{x}_i^t)$  is the likelihood that a beam ending at position  $\langle x, y \rangle$  is reflected by person  $i$ , given the state  $\mathbf{x}_i^t$  of that person. To compute this quantity, we construct a two-dimensional and normalized histogram by counting how many samples representing the belief about  $\mathbf{x}_i^t$  fall into each bin.

Now suppose  $x_b$  and  $y_b$  are the coordinates of the cell in which the beam  $b$  ends. Accordingly, we can compute the probability  $P(\text{hit}_b | \mathbf{x}_i^t)$  that a beam  $b$  is reflected by a person as

$$P(\text{hit}_b | \mathbf{X}^t) = P(\text{hit}_{x_b, y_b} | \mathbf{X}^t). \quad (11)$$

It remains to describe, how we incorporate the quantity  $h_b = P(\text{hit}_b | \mathbf{X}^t)$  into the scan alignment and registration process. If we consider all beams as independent, the likelihood  $p(s_t | l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1}))$  of the most recent measurement give all previous scans is obtained as:

$$p(s_t | l_t, \hat{m}(\hat{l}^{t-1}, s^{t-1})) = \prod_{b \in s_t} p(b | \hat{m}(\hat{l}^{t-1}, s^{t-1}))^{(1-h_b)}. \quad (12)$$

Thus, during the scan alignment we weigh each beam  $b$  according to the probability  $1 - P(\text{hit}_b \mid \mathbf{X}^t)$ . Please note that this is a general form of a situation in which it is exactly known whether or not  $b$  is reflected by a person. If  $b$  is known to be reflected by a person,  $h_b$  equals 1 such that  $b$  does not change the likelihood of the scan (see Figs 6 and 7).

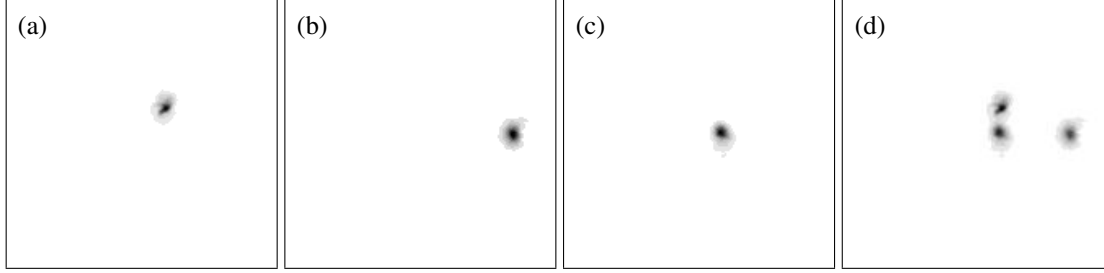


Figure 6. Example situation in which the quantity  $P(\text{hit}_{x,y} \mid \mathbf{X}^t)$  (image d) is computed by combining the histograms for three individual trackers (image a-c).

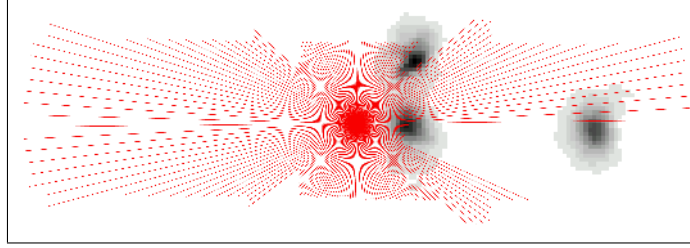


Figure 7. The weight of a laser beam is computed according to the value  $P(\text{hit}_{x,y} \mid \mathbf{X}^t)$  of the cell in which it ends.

The second task is to filter out beams reflected by persons to avoid spurious objects in the resulting maps. In our current system we compute a bounding box for each sample set  $\mathbf{S}_i^t$  and integrate only those beams whose endpoint does not lie in any of the bounding boxes. To cope with the possible time delay of the trackers, we also ignore corresponding beams of several previous and next scans before and after the person was detected. Please note, that one generally can be more conservative during the map generation process, because the robot generally scans every part of the environment quite often. However, during scan alignment, a too conservative strategy may result in too few remaining beams which leads to reduced accuracy of the estimated positions.

## 5 Experiments

The approach described above has been implemented and tested on different robotic platforms and based on extensive off-line experiments carried out with recorded data. The goal of the experiments described in this section is to illustrate that the integration of people detection techniques into the mapping process leads to better maps since the resulting alignments are more accurate and since beams reflected by persons are filtered out which reduces the number of spurious objects. Please note that our current implementation can track several people in real-time, so that the time to map an environment is not influenced by using this information.

### 5.1 Learning 2d Maps

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. Figure 8 shows the robot during the mapping process. There were 15 people walking through the environment while the robot was mapping it. The map obtained without filtering measurements reflected by persons is shown in the left image of Figure 9. The result obtained with our new algorithm is shown in the right image of Figure 9. Both maps are high-resolution



Figure 8. Robot Sam mapping the populated exhibition hall of the Byzantine Museum in Athens.

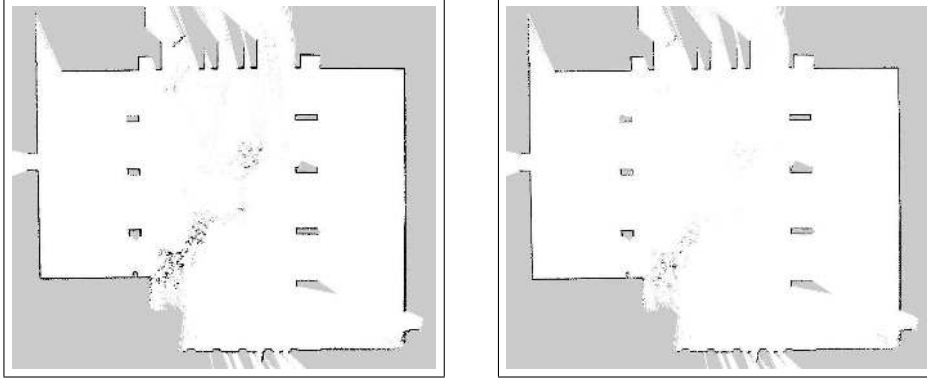


Figure 9. Maps of the Byzantine Museum in Athens created without (left image) and with people filtering (right image).

occupancy grid maps with a resolution of 2cm per cell. As can be seen from the figures the number of spurious readings is reduced considerably. The remaining spurious objects come from a crowd of people that did not move during the mapping process. Accordingly, they could not be filtered out by our algorithm. Furthermore, it should be mentioned, that the museum hall contains several columns which produce similar features in the range scans as people. Nevertheless, our approach could seriously reduce the number of readings corrupted by people.



Figure 10. Robot Rhino mapping the populated corridor environment at the University of Bonn.

We carried out a second experiment using our RWI B21 robot Rhino in a 25m x 4m large corridor environment of the Computer Science Department in Bonn. Again, while the robot was gathering the data, there were several (up to five) persons walking through the environment (see Figure 10). The map obtained without people filtering is shown in the left image of Figures 11. As can be seen from the figure, there are a lot of cells in the resulting grid map, which have a high occupancy probability since people covered the corresponding area while the robot was mapping the environment. If, however, we use our new algorithm and filter out the beams corresponding to persons, the effect of the persons is seriously reduced in the resulting map (see right image of Figures 11). In this experiment the overall reduction of beams reflected by the persons is 97.5%.

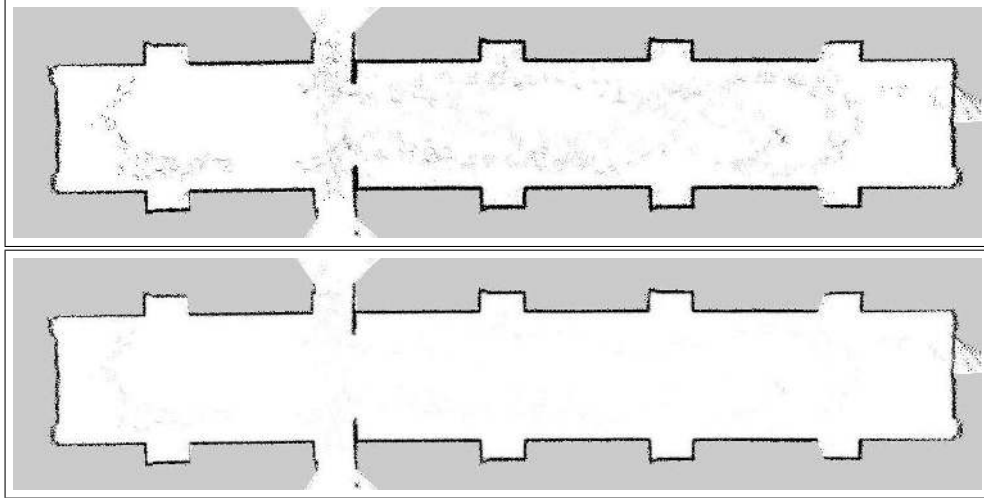


Figure 11. Occupancy grid maps created for the populated corridor environment of the University of Bonn without (upper image) and with people filtering (lower image).

## 5.2 Improved Robustness

Besides the fact that the resulting maps are better, filtering people increases the robustness of the mapping process. To demonstrate this we have carried out a series of experiments in which we added random noise to the poses in the input data and compared the performance of our mapping strategy with and without people filtering. We performed 50 experiments for each noise level. Figure 12 shows the numbers of maps containing a translational error larger than 200cm for the different noise values. In this figure the x-axis corresponds to the standard deviation of the Gaussian noise added to each odometry reading. The y-axis is the number of maps in which the translational error after registration exceeds 200cm. As can be seen by the figure, the use of the information provided by the people tracker significantly increases the accuracy of the position estimation during the mapping process.

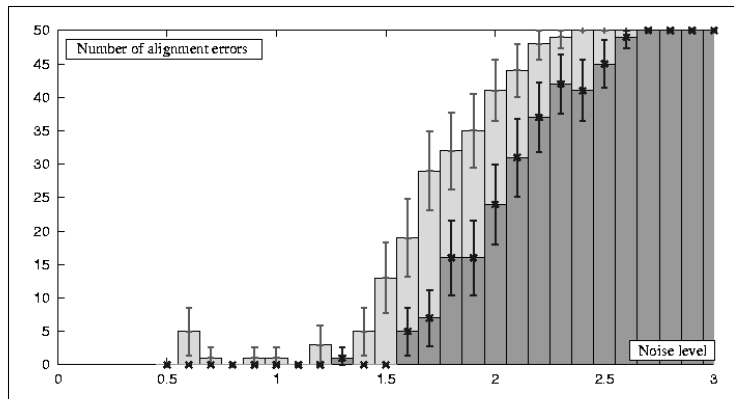


Figure 12. Number of maps with translational error larger than 2m computed without people filtering (light grey) and using people filtering (dark grey) for increasing levels of noise in odometry.

## 5.3 Learning 3d Maps

The last experiment was carried out to analyze the performance of our system when learning three-dimensional maps. For this experiment we used the Pioneer 2 AT platform (see Figure 13 (left)) equipped with two laser



Figure 13. Pioneer 2 AT robot Herbert for 3d outdoor mapping (left) and typical situation in which people walk through the scene during mapping (right).



Figure 14. Spurious objects caused by people walking through the environment.

range-scanners. Whereas the first scanner, that is mounted in front of the robot, is used for tracking people, the second scanner, that is mounted on an AMTEC wrist module, is used to scan the 3d structure of the environment. Figure 13 (right) shows a typical scenario during this experiment performed on our university campus. Here, several people were walking through the scene while the robot was scanning it. Figure 15 (left) depicts the model obtained after aligning two scans of the same environment. In this model, the people appear as three-dimensional curves. Figure 14 contains a magnified view of the corresponding portion of the map. If we integrate the information obtained from the people tracker, however, these spurious objects are completely removed (see right image of Figure 15). The number of triangles in these models are 416.800 without filtering and 412.500 with filtering. Please note, that this experiment also illustrates the advantage of using a tracking system over a pure feature-based approach. Due to the displacement of the scanners, people are not always visible in both scanners. Accordingly, a purely feature-based approach like [33] will add objects to the 3d model whenever they are not detected by the first scanner. Our system, however, can predict positions of persons in the case of occlusions and thus can filter out the corresponding readings even if the features are missing.

## 6 Conclusions

In this paper we presented a probabilistic approach to mapping in populated environments. The key idea of this technique is to use Sample-based Joint Probabilistic Data Association Filters (SJPDFAs) to track people in the data obtained with the sensors of the robot. The results of the people tracking are integrated into the scan alignment process and into the map generation process. This leads to two different improvements. First, the resulting pose estimates are better and second, the resulting maps contain less spurious objects than the maps created without

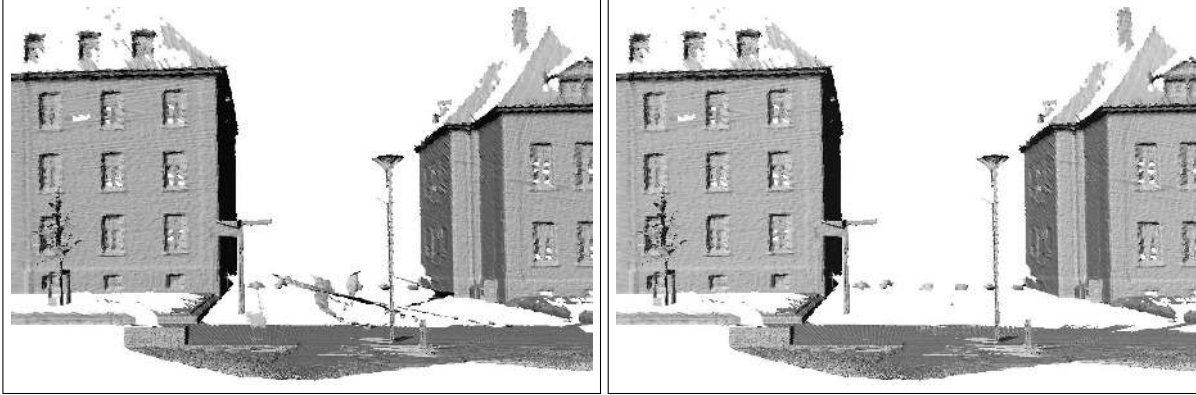


Figure 15. Three-dimensional map of a building (left) and people filtered (right).

filtering people.

Our technique has been implemented and tested on different robotic platforms as well as for generating 2d and 3d maps. The experiments demonstrate that our approach can seriously reduce the number of beams corrupted by people walking through the environment. Additionally, extensive simulation experiments illustrate that the pose estimates are significantly better if the results of the tracking system are incorporated during the pose estimation.

## Acknowledgments

This work has partly been supported by the EC under contract number IST-2000-29456.

## References

- [1] K.O. Arras, R. Philippsen, M. de Battista, M. Schilt, and R. Siegwart. A navigation framework for multiple mobile robots and its applications at the Expo.02 exhibition. In *Proc. of the IROS-2002 Workshop on Robots in Exhibitions*, 2002.
- [2] M. Bennewitz, W. Burgard, and S. Thrun. Using EM to learn motion behaviors of persons with mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [3] P. Besl and N. McKay. A method for registration of 3d shapes. *Trans. Patt. Anal. Mach. Intell.* 14(2), pages 239–256, 1992.
- [4] D. Beymer and Konolige K. Tracking people from a mobile platform. In *IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics*, 2001.
- [5] H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model. *Intl. J. of Pattern Rec. and AI*, 2001.
- [6] 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.
- [7] 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.
- [8] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.
- [9] 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.

- [10] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Transactions on Robotics and Automation*, 3(3):249–265, 1987.
- [11] 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.
- [12] M.A. Greenspan and G. Godin. A nearest neighbor method for efficient ICP. In *Proc. of the 3rd Int. Conf. on 3-D Digital Imaging and Modeling (3DIM01)*, 2001.
- [13] 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.
- [14] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3d models of indoor and outdoor environments with a mobile robot. In *Proc. of the European workshop on advanced mobile robots (EUROBOT)*, 2001.
- [15] 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.
- [16] Roger E. Kahn, Michael J. Swain, Peter N. Prokopowicz, and R. James Firby. Gesture recognition using the perseus architecture. Technical Report TR-96-04, University of Chicago, 19, 1996.
- [17] B. Kluge, C. Koehler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.
- [18] D. Kortenkamp, E. Huber, and R. P. Bonasso. Recognizing and interpreting gestures on a mobile robot. In *Proc. of the American Conference on Artificial Intelligence*, 1996.
- [19] E. Kruse and F. Wahl. Camera-based monitoring system for mobile robot guidance. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.
- [20] S. M. Lavalle, H. H. Gonzalez-Banos, G. Becker, and J.-C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1997.
- [21] 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.
- [22] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
- [23] 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.
- [24] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.
- [25] H. Samet. *Applications of Spatial Data Structures*. Addison-Wesley Publishing Company, 1990.
- [26] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving objects with a mobile robot. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [27] H. Shatkay. *Learning Models for Robot Navigation*. PhD thesis, Computer Science Department, Brown University, Providence, RI, 1998.
- [28] C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [29] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori. On motion planning of mobile robots which coexist and cooperate with human. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1995.

- [30] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.
- [31] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.
- [32] S. Waldherr, S. Thrun, R. Romero, and D. Margaritis. Template-based recognition of pose and motion gestures on a mobile robot. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.
- [33] 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.
- [34] Q. Zhu. Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3), 1991.