

# Probabilistic state estimation of dynamic objects with a moving mobile robot

Dirk Schulz<sup>a,\*</sup>, Wolfram Burgard<sup>b</sup>

<sup>a</sup> Department of Computer Science III, University of Bonn, 53117 Bonn, Germany

<sup>b</sup> Department of Computer Science, University of Freiburg, D-79110 Freiburg, Germany

## Abstract

Mobile service robots are designed to operate in dynamic and populated environments. To plan their missions and to perform them successfully, mobile robots need to keep track of relevant changes in the environment. For example, office delivery or cleaning robots must be able to estimate the state of doors or the position of waste-baskets in order to deal with the dynamics of the environment. In this paper we present a probabilistic technique for estimating the state of dynamic objects in the environment of a mobile robot. Our method matches real sensor measurements against expected measurements obtained by a sensor simulation to efficiently and accurately identify the most likely state of each object even if the robot is in motion. The probabilistic approach allows us to incorporate the robot's uncertainty in its position into the state estimation process. The method has been implemented and tested on a real robot. We present different examples illustrating the efficiency and robustness of our approach. © 2001 Elsevier Science B.V. All rights reserved.

*Keywords:* Probabilistic state estimation; Sensor interpretation; Mobile service robots

## 1. Introduction

Mobile service robots are designed to perform complex tasks in populated environments. In this context, the position and orientation of certain objects in the environment generally has a serious influence on the overall performance of the system. For example, doors can be closed and tables, chairs or waste-baskets can block the path of the robot and thus prevent it from executing a given plan. To robustly perform their tasks mobile robots therefore must be able to keep track of the state of objects which are relevant for the successful execution of their missions.

In this article we propose a technique allowing a mobile robot to continuously estimate the state of dynamic objects based on the data obtained with its

sensors. Using this approach enables a robot to update and maintain its knowledge about the state of the environment while it is executing its task. Based on this information it can immediately react to changes of the environment and e.g. improve its plans dynamically. To robustly deal with the different types of uncertainty and noise we apply a probabilistic approach and maintain probability densities over the possible states of the dynamic objects. Additionally, we use a variant of Markov localization to deal with the robot's uncertainty in its position while it moves through the environment. Our approach uses sample-based representations of the different densities and includes techniques allowing their efficient integration. In this paper we also present an application of the method in which a mobile robot maintains an object-based geometric world model over time using its laser-range sensor. The Bayesian integration step of the object state estimator is based on comparing the measurements

\* Corresponding author.

E-mail address: schultz@tuor.informatik.uni-bonn.de (D. Schulz).

obtained with the robot's sensors with the expected sensor measurements given a 3D-boundary representation of the environment and the robot's current belief.

The problem of tracking or locating objects from a moving robot has recently been investigated at different sites. For example, Baerfeldt [1] describes a method to identify and localize objects in single images. Additionally, Margaritis and Thrun [15] and Sahin and Gaudio [22] present methods to localize one or several objects from sequences of camera images. Both approaches use models of the robots kinematic to deal with the noise in the robot's odometry. In [17] the condensation algorithm is used to keep track of multiple objects with a moving robot. As well as in [15] this method explicitly models the uncertainty in the position of the objects. All these approaches do not explicitly take into account the robots uncertainty in its position when tracking the object. They only increase the uncertainty of the estimate according to the movements carried out by the robots. Our approach also considers the robot's belief about its position when estimating the state of objects. This way it can more accurately deal with the movements carried out by the robot between consecutive measurements.

The problem to acquire and maintain a model of the environment has been a major research area in mobile robotics. The most frequently used types of models are metric and topological maps. Topological models, as used in [14,20], describe the environment at a coarse resolution and in a graph-like structure. Because such models lack important details such as doors and tables, they are only of limited use for estimating the state of dynamic objects. Metric maps, on the other hand, describe the environment at a finer level of detail. A popular approach is discrete occupancy grids proposed in [7,18]. Each cell of such a grid contains the probability that the corresponding space in the environment is occupied. The major advantage of occupancy maps lies in the existence of techniques for their acquisition and maintenance based on sensor information. However, most of the occupancy grid techniques are intended for static environments only. Since all cells are considered independently, they cannot appropriately represent dynamic objects. In this paper we use an object-based model of the environment containing all relevant objects of the environment. We apply a probabilistic approach to estimate the state of dynamic objects. Changes of these states are also updated in the

object-based model so that the robot's planning components can quickly adapt the robot's course of action to the new situation.

The remainder of this paper is organized as follows. After introducing the state estimation technique in the next section, Section 3 describes the application of this technique using laser-range measurements. In Section 5 we present several experiments carried out with our mobile robot Rhino in our office environment. They demonstrate the efficiency and robustness of the overall approach.

## 2. Bayesian state estimation

### 2.1. Object state estimation

In our approach we assume that the changes in the environment come from certain objects whose state is changed by the people living in the environment. Typical examples are doors which are opened or closed or chairs and tables, the configuration of which is frequently changed because they are used by the inmates.

A robot moving through such an environment in principle has to simultaneously maintain the joint distribution over all states of the dynamic objects within the environment and its own location. Unfortunately, this approach is not tractable since the size of the joint state space grows exponentially with the number of objects. Therefore, we consider a marginal distribution and independently estimate the states of the dynamic objects and of the robot.

Within the Bayesian framework a probability density of an object's state  $s \in S$  is maintained conditioned on the observations  $o$  (sensor measurements) obtained. The well-known Bayesian update formula is applied to determine the new posterior  $p(s|o)$  whenever a new observation is obtained:

$$p(s|o) = \frac{p(o|s) \cdot p(s)}{p(o)}. \quad (1)$$

The term  $p(s)$  is the prior density of the object's state. It is generally derived from the initial belief of the object's state and the previous measurements by the recursive application of Eq. (1).

In our special case of determining the state of objects like the opening angle of doors using the sensors of the robot, we have to take the robot's un-

certainty about its current position into account. This is achieved by integrating over all possible locations  $l$  of the robot during the Bayesian update step:

$$p(s|o) = \int_L \frac{p(o|s, l) \cdot p(s|l) \cdot p(l)}{p(o|l)} dl. \quad (2)$$

Under the strong assumption that the state  $s$  of the object is independent from the location  $l$  of the robot, we obtain

$$p(s|o) = \int_L \frac{p(o|s, l) \cdot p(s) \cdot p(l)}{p(o|l)} dl. \quad (3)$$

The denominator  $p(o|l)$  can be rewritten as

$$p(o|l) = \int_S p(o|s, l) \cdot p(s) ds. \quad (4)$$

The state of a dynamic object changes over time. Therefore, the belief  $p(s)$  of the object's state has to be updated according to a model of the object's dynamics  $p(s|t, s')$  prior to a state estimate, where  $t$  is the time elapsed since the previous estimate.

$$p(s) = \int_S p(s|t, s') \cdot p(s') ds'. \quad (5)$$

## 2.2. Robot position estimation using Markov localization

To estimate the position  $l \in L$  of the robot in its environment, we also apply a Bayesian filtering technique also denoted as *Markov localization* [9] which has successfully been applied in a variety of successful robot systems [2,27]. The key idea of Markov localization is to maintain the probability density of the robot's own location  $p(l)$ . Markov localization also uses the recursive Bayesian update formula to update the  $p(l)$  about the robot's position whenever a new sensor measurement is received:

$$p(l|o) = \alpha \cdot p(o|l) \cdot p(l). \quad (6)$$

Here  $\alpha$  is a normalize ensuring that the  $p(l|o)$  sum up to 1 over all  $l$ . The term  $p(o|l)$  denotes the probability of making observation  $o$  given that the robot's current location is  $l$ . It highly depends on the information the robot possesses about the environment and the sensors used. Different kinds of realizations can be found in [2,12,13,19].

Additionally, Markov localization uses a well-known formula coming from the domain of Markov chains to update the belief  $p(l)$  whenever the robot performs a movement action  $a$ :

$$p(l|a) = \int p(l|a, l') \cdot p(l') dl'. \quad (7)$$

In this equation the term  $p(l|a, l')$  describes the probability that the robot is at position  $l$  given that it executed the movement  $a$  at position  $l'$ .

This completes the derivation of the necessary equations for our object state estimation procedure. The whole process to estimate the states of dynamic objects from a moving robot is summarized in Table 1. Please note that, since we consider the position state of the robot independent from the states of the individual objects, we have to use different sensor readings for the robot localization and for object state estimation.

## 2.3. Sample-based realization

In the previous section we left open how to represent the beliefs  $p(l)$  and  $p(s)$  of the robot's position and the state of the object. Over the past years, different techniques have been used to represent the beliefs. Among them are piecewise constant approximations as applied in [3,4,12,15,19,25]. A very popular approach is to use Gaussians [10,16,24,26] to represent the densities. In this paper we use particle-filters to approximate the involved densities. The key idea of particle filters is to use a sample-based representation for the densities. The updates are carried out using resampling techniques (see e.g. [6,11,21]).

To estimate the position of the robot we apply a variant of Markov localization denoted as Monte Carlo localization [5,8]. In robot localization, the two steps presented in the previous section are realized by the following two procedures:

1. *Prediction.* In the prediction step, each sample is updated according to the model  $p(l|a, l')$  of the robot's dynamics and the action  $a$  executed since the previous update.
2. *Correction.* In the correction step, the new observation  $o$  is integrated into the sample set. This is done by bootstrap resampling, where each sample is weighted according to the likelihood  $p(o|l)$  of

Table 1  
Simultaneous object state estimation and localization for a moving robot

---

```

for each object  $i$  do
  generate the samples representing the prior  $p(s_i)$ 
end for

 $k \leftarrow 0$ 
forever do
  if the robot receives new sensory input  $o_k$  do
    if even ( $k$ )
      /* integrate  $o_k$  into  $p(l)$  */
      for each location  $l$  do
         $p(l|o_k) \leftarrow \alpha p(o_k|l)p(l)$ 
      end for
    else
      for each object  $i$  do
        /* integrate  $o_k$  into  $p(s_i)$  */
         $p(s_i) \leftarrow \int_{\mathcal{S}} p(s_i|t, s'_i) \cdot p(s'_i) ds'_i$ 
         $p(s_i|o_k) \leftarrow \int_{\mathcal{L}} (p(o_k|s_i, l) \cdot p(s_i) \cdot p(l)) / p(o_k|l) dl$ 
      end for
    end if
     $k \leftarrow k + 1$ 
  else if the robot receives a new odometry reading  $a$  do
    for each location  $l$  do
      /* integrate  $a$  into  $p(l)$  */
       $p(l) \leftarrow \int_{\mathcal{L}} p(l|a, l')p(l') dl'$ 
    end for
  end if
end forever

```

---

making observation  $o$  given sample  $l$  is the current state of the system.

Almost the same method is applied during the state estimates of dynamic object's; using a sample-based representation of the densities, Eqs. (3) and (4) simplify to

$$p(s|o) = \sum_{l \in \mathcal{L}} \frac{p(o|s, l) \cdot p(s)}{p(o|l)}, \quad (8)$$

$$p(o|l) = \sum_{s \in \mathcal{S}} p(o|s, l). \quad (9)$$

Here,  $\mathcal{L}$  and  $\mathcal{S}$  denote the sample sets representing the prior densities  $p(l)$  and  $p(s)$ .

According to these equations all we need to know are the quantities  $p(s)$  which is the prior distribution of the state  $s$  of the currently considered object, the current belief  $p(l)$  of the position of the robot as well as the term  $p(o|s, l)$ . This term is the crucial part of the state estimation process, since it describes the likelihood of making a certain observation  $o$  given the state  $s$  of an object and the location  $l$  of the robot. This quantity is generally derived using a matching procedure which determines the likelihood

of the current measurement given the measurement that is expected for a given position of the robot and the state of the object. A special implementation of this scheme using laser-range data is described in the next section.

### 3. Laser-based implementation

Our current implementation uses the data obtained from laser-range finders to estimate the state of dynamic objects such as doors, chairs and tables. Each scan of such a laser-range device measures the distances to objects along beams at a fine angular resolution (see, for example, the left image of Fig. 1). The result is a sequence of  $n$  distance measurements  $d_i$ , where  $n$  is the number of beams and  $d_i$  is the length of beam  $i$ . To determine the likelihood  $p(o|s, l)$  of a beam we use a geometric world model consisting of a 2D/3D-boundary representation of our environment (see right image of Fig. 2). Given the geometric primitives in this map we use a sensor simulation based on ray-tracing to compute for each beam  $i$  the expected distance  $\hat{d}_i(s, l)$ , given the robot is at loca-

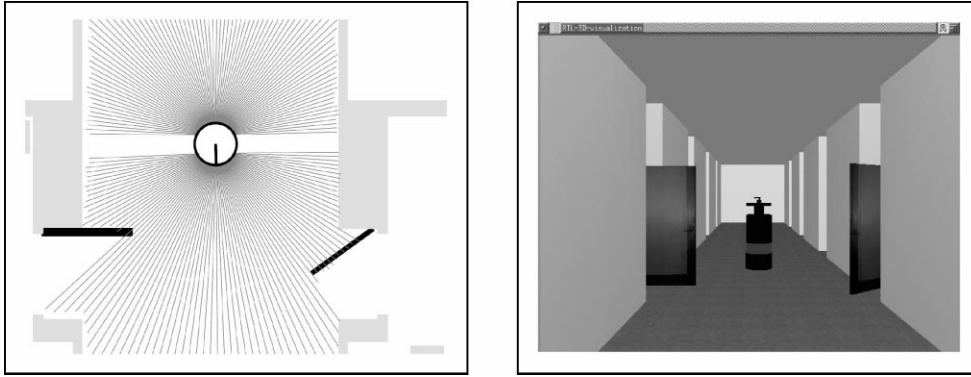


Fig. 1. Left image: Rhino detecting the state of two doors with its laser range-finder. The estimated states of the doors are depicted in black. Right image: 3D visualization of the situation depicted in the left image.

tion  $l$  and the state of the object is  $s$ . Obviously, the likelihood  $p(o|s, l)$  highly depends on the accuracy of the sensors and the map. In our current system we assume that the measurement error is Gaussian, i.e.  $d_i$  is normally distributed around the expected distance  $\hat{d}_i(s, l)$ .

Suppose  $p(d_i|\hat{d}_i(s, l))$  is the probability of measuring a distance  $d_i$  with the  $i$ th beam, given the distance to the closest object in the map is  $\hat{d}_i(s, l)$ . Under the assumption that the individual beams are independent, given the position of the robot and the map, we can compute  $p(o|s, l)$  as

$$p(o|s, l) = \prod_{i=1, \dots, n} p(d_i|\hat{d}_i(s, l)). \quad (10)$$

To speed up ray-tracing, we maintain a spatial index structure on the world model, which is based on spatial tiling [23] of the scene. In addition, in most cases it is not necessary to consider all beams of one scan for a state estimate. Beams, which do not hit the object in any of the states contained in the sample set, only provide a constant factor in the probability calculation (see Eq. (10)). Based on the sample set of the prior distribution  $p(s)$ , the sequence of beams being used can therefore be reduced to the beams hitting the objects in at least one of the states contained in the sample set. We approximate this reduced sequence by computing the set of consecutive beams which intersect the bounding box of the object in at least one of the sample states.

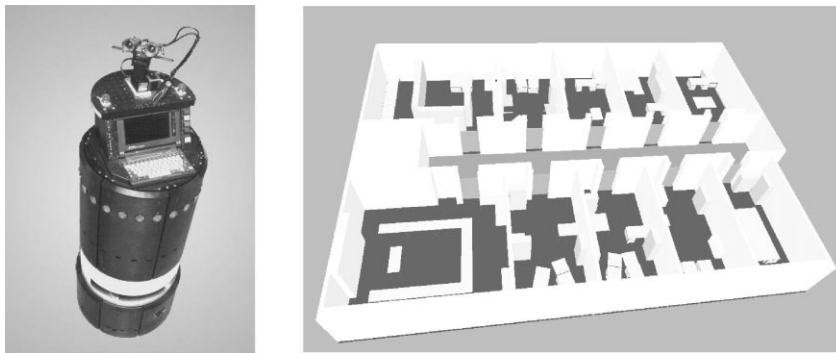


Fig. 2. Left image: the RWI B21 robot Rhino. Right image: a VRML visualization of the 3D model of our office environment used for the experiments.

#### 4. Experimental results

In order to evaluate the performance of the state estimation technique, we carried out several experiments with our mobile robot Rhino in our office environment. Rhino is an RWI B21 robot (see Fig. 2) equipped with two SICK PLS 200 laser-range finders. These sensors measure the distance to obstacles in the surrounding of the robot with an angular resolution of  $1^\circ$ , thus providing 360 distance measurements in each scan. The estimation tasks carried out by the robot during the experiments were:

1. to determine the state of doors while it is moving fast,
2. to determine the position and orientation of a single object while passing by, and
3. to correct the position of furniture within the environment.

All experiments have been evaluated off-line based on logs recorded during the robot's operation. These

logs contain the sequence of laser measurements, as well as the robot's belief of its own position. This belief is given as a set of position samples, which is provided by the robot's Markov localization component. The evaluation of the state estimation algorithm was carried out on a SUN Ultra-Sparc 5 workstation.

##### 4.1. Estimating the state of doors

In order to demonstrate that our state estimation technique is capable to provide robust and accurate estimates even when the robot is moving fast, we estimated the state of the doors in our department building (see Fig. 2 for the complete model of the department), while the robot was moving along the corridor. Fig. 3 shows a bird's eye view of the corridor, the state of the doors during the experiments, and the trajectory taken by Rhino (see also Fig. 4). At the straight parts of the trajectory the robot moved at a speed of 70 cm/s. A door's state is estimated by the robot, as soon as it gets closer than 3.5 m. We approximate the opening angle

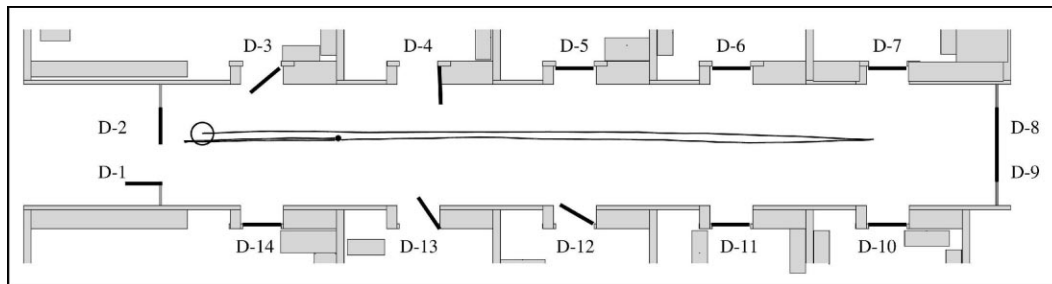


Fig. 3. Trajectory of the robot and state of the doors in the corridor of the Department of Computer Science, Bonn.

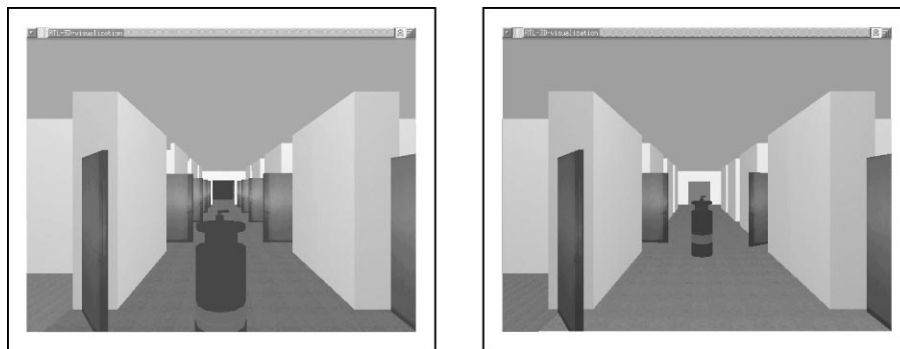


Fig. 4. Visualization of the corridor at the beginning (left image) and at the end of the trajectory (right image).

Door	D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	D-9	D-10	D-11	D-12	D-13	D-14
avg. state	180	270	220	264	182	180	182	270	90	176	178	155	124	173
std. dev.	0.0	0.0	5.3	6.4	5.0	0.0	4.3	0.0	0.0	4.6	3.9	5.0	7.1	6.5
measurements	21	25	43	23	17	17	33	27	26	37	13	15	19	41
exact values	180	270	218	270	180	180	180	270	90	180	180	150	122	180

Fig. 5. Estimated states of the doors depicted in Fig. 3; given are the means (average state) of a number of estimates (measurements) for each door, as well as the standard deviations; the state denotes the orientation of the door within the world model in degree.

in steps of  $10^\circ$  ranging from  $0^\circ$  for a completely closed door, to  $90^\circ$  for a completely opened door. Using this coarse approximation, one state estimate takes about 250 ms. Note that in some locations on the corridor, the robot has to measure up to 4 doors.

Fig. 5 lists the averages of independent estimates for each door where the states are described as the orientation of the door within the world model. On average over all doors, the estimate is only half a state apart from the exact value. The algorithm is thus capable to collect object state information even when the robot moves at high speed.

#### 4.2. Estimating the state of a movable object

The second experiment is to estimate the position and orientation of objects which might have been moved since the last time the robot observed them. Fig. 6 shows the trajectory of Rhino while it is inspecting the state of an U-shaped object on the corridor. In this experiment, the object was manually rotated by about  $30^\circ$  after the robot had passed it two times. This

way, three different states of the object have been presented to the robot. In contrast to estimating the state of doors, this estimation task has three degrees of freedom. In this situation, the efficiency of the state estimation process strongly depends on the prior information of the object's state. In the experiment, we assume that the position of the object is roughly known from the previous estimate. The estimator used a sample set of 40 states. The algorithm was able to initially find the object and to track it using this setting. Fig. 7 lists the average estimate for each pose and the standard deviations thereof. In our prototypic implementation, an estimate took between 1 and 3 s, depending on how many laser beams hit the object.

#### 4.3. Maintaining the world model

In the final experiment, we tested the algorithm on the object-based model of our office environment. The task is to estimate the position of 4 desks located in two offices. For the experiment we modified the model by shifting the desks 20 cm towards the center of the

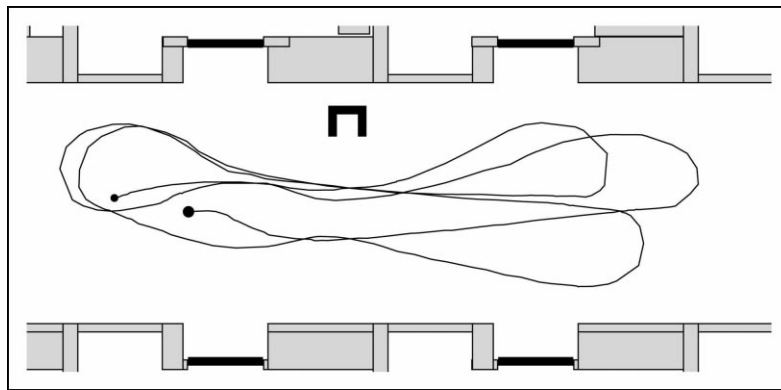


Fig. 6. RHINO estimates the position and orientation of an U-shaped object placed on the corridor; the object is 37 cm by 37 cm in size.

State	sample	measurements	x	y	$\theta$	std. dev. $l$	std. dev. $\theta$
1	40	108	1299.7	1088.6	0.01	12.5	0.13
2	40	100	1297.2	1081.9	-0.84	14.7	0.13
3	40	61	1290.1	1088.6	-1.24	10.8	0.09

Fig. 7. State estimates of an U-shaped obstacle on the corridor (see Fig. 6).

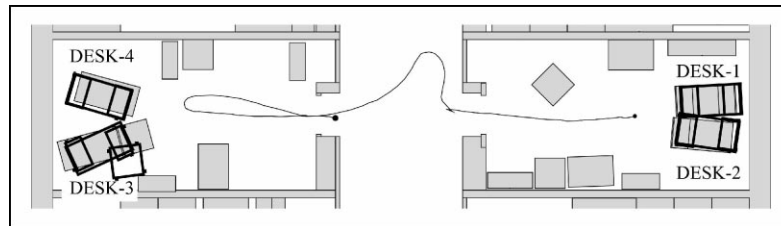


Fig. 8. RHINO estimating the position of desks in two offices; the robot started in the doorway of the left office.

rooms. The situation is illustrated in Fig. 8. The initial state of the model is drawn in grey. Our algorithm managed to correctly identify the positions of Desk-1, Desk-2 and Desk-4, as can be seen from the estimated states outlined in black. However, it was not able to determine the state of Desk-3 correctly. Desk-3 was partially occluded by a small square table standing in front of it and additionally by a chair, which was not contained in the model. The algorithm also failed to correctly determine the position of the square table, as it is nearly invisible to the laser-range finders.

## 5. Conclusions

In this work, we proposed the probabilistic state estimation of dynamic objects. The algorithm achieves a high degree of robustness by taking the robot's uncertainty about its position into account. Therefore, a Markov localization component is integrated into the state estimation process. The state estimator compares real sensor measurements against expected measurements given a state of the object, where the expected measurements are obtained using a sensor simulation within an object-based model of the environment. The use of Monte Carlo sampling techniques throughout the algorithm ensures that the robot is capable to determine the state of objects very fast. As our experiments show, the approach allows for efficient and

robust state estimates even when the robot is moving at speeds of upto 70 cm/s.

Even though these results are promising, there are still warrants for future work. Methods to detect poor estimates caused for example by unexpected occlusions can further increase the robustness of the algorithm. In certain situations, especially if the objects are almost symmetric, a state estimate based on one measurement may not be sufficient to determine the most likely state of the object uniquely. To solve this problem, the approach needs to be extended to use sequences of measurements taken from different positions. That way, the accuracy of state estimates might also be increased. In order to actively maintain a world model, a planning component needs to be developed, which decides on the optimal position of the robot for a state estimate, e.g. the position which provides the sensor information that is expected to achieve the state estimate with the minimal variance of the posterior object state density.

## References

- [1] A.-J. Baerveldt, A vision system for object verification and localization based on local features, in: *Proceedings of the Third European Workshop on Advanced Mobile Robots (EUROBOT)*, 1999, pp. 57–64.
- [2] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, Experiences with an interactive museum tour-guide robot, *Artificial Intelligence* 114 (1–2) (1999) 3–55.



- [3] W. Burgard, A. Derr, D. Fox, A.B. Cremers, Integrating global position estimation and position tracking for mobile robots: The dynamic Markov localization approach, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1998.
- [4] W. Burgard, D. Fox, D. Hennig, T. Schmidt, Estimating the absolute position of a mobile robot using position probability grids, in: Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-96), Portland, OR, 1996, pp. 896–901.
- [5] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte Carlo localization for mobile robots, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Detroit, MI, 1999.
- [6] A. Doucet, On sequential simulation-based methods for Bayesian filtering, Technical Report CUED/F-INFENG/TR 310, Department of Engineering, University of Cambridge, Cambridge, UK, 1998.
- [7] A. Elfes, Sonar-based real-world mapping and navigation, IEEE Transactions on Robotics and Automation 3 (3) (1987) 249–265.
- [8] D. Fox, W. Burgard, F. Dellaert, S. Thrun, Monte Carlo localization: Efficient position estimation for mobile robots, in: Proceedings of the National Conference on Artificial Intelligence (AAAI-99), Orlando, FL, 1999.
- [9] D. Fox, W. Burgard, S. Thrun, Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research 11 (1999) 391–427.
- [10] J.-S. Gutmann, C. Schlegel, AMOS: Comparison of scan matching approaches for self-localization in indoor environments, in: Proceedings of the First Euromicro Workshop on Advanced Mobile Robots, IEEE Computer Society Press, Silver Spring, MD, 1996.
- [11] M. Isard, A. Blake, Contour tracking by stochastic propagation of conditional density, in: Proceedings of the European Conference of Computer Vision, 1996.
- [12] L.P. Kaelbling, A.R. Cassandra, J.A. Kurien, Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1996.
- [13] K. Konolige, Markov localization using correlation, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99), Stockholm, Sweden, 1999.
- [14] D. Kortenkamp, T. Weymouth, Topological mapping for mobile robots using a combination of sonar and vision sensing, in: Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94), Seattle, WA, 1994, pp. 979–984.
- [15] D. Margaritis, S. Thrun, Learning to locate an object in 3D space from a sequence of camera images, in: Proceedings of the International Conference on Machine Learning (ICML), 1998.
- [16] P.S. Maybeck, The Kalman filter: An introduction to concepts, in: I.J. Cox, G.T. Wilfong (Eds.), Autonomous Robot Vehicles, Springer, New York, 1990, pp. 194–204.
- [17] E.B. Meier, F. Ade, Using the condensation algorithm to implement tracking for mobile robots, in: Proceedings of the Third European Workshop on Advanced Mobile Robots (EUROBOT), IEEE, 1999, pp. 73–80.
- [18] H.P. Moravec, Sensor fusion in certainty grids for mobile robots, AI Magazine 9 (2) (1988) 61–74.
- [19] I. Nourbakhsh, R. Powers, S. Birchfield, DERVISH an office-navigating robot, AI Magazine 16 (2) (1995) 53–60.
- [20] D. Pierce, B. Kuipers, Learning to explore and build maps, in: Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94), Seattle, WA, 1994.
- [21] D.B. Rubin, Using the SIR algorithm to simulate posterior distributions, in: M.H. Bernardo, K.M. DeGroot, D.V. Lindley, A.F.M. Smith (Eds.), Bayesian Statistics, Vol. 3, Oxford University Press, Oxford, UK, 1988.
- [22] E. Sahin, P. Gaudiano, Development of a visual object localization module for mobile robots, in: Proceedings of the Third European Workshop on Advanced Mobile Robots (EUROBOT), 1999, pp. 65–72.
- [23] H. Samet, Applications of Spatial Data Structures, Addison-Wesley, Reading, MA, 1990.
- [24] B. Schiele, J.L. Crowley, A comparison of position estimation techniques using occupancy grids, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), San Diego, CA, 1994.
- [25] R. Simmons, S. Koenig, Probabilistic robot navigation in partially observable environments, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Que., 1995.
- [26] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics, in: I. Cox, G. Wilfong (Eds.), Autonomous Robot Vehicles, Springer, Berlin, 1990, pp. 167–193.
- [27] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, D. Schulz, MINERVA: A second generation mobile tour-guide robot, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Orlando, FL, 1999.



**Dirk Schulz** is a Ph.D. student at the Department of Computer Science III at the University of Bonn. His main research interests are in the fields of service robotics, probabilistic state estimation and Internet-based telepresence.



**Wolfram Burgard** is an Associate Professor for computer science at the University of Freiburg and head of the research lab for Autonomous Intelligent Systems. His areas of interest lie in artificial intelligence and mobile robotics.