State Estimation Techniques for 3D Visualizations of Web-based Tele-operated Mobile Robots

Dirk Schulz, Wolfram Burgard, Armin B. Cremers

The World Wide Web provides a unique opportunity to connect robots to the Internet. Easy to use Web interfaces enable people all over the world to control mobile robots and to monitor their operation from the distance. In this context visualizing the robot's actions plays a key role. This paper presents probabilistic state estimation techniques for a system which provides smooth real-time 3D-visualizations of a mobile robot within a dynamic environment over the Internet. Using the robot's sensors, the state estimators continuously determine the position and the pose of dynamic objects which influence the behavior of the robot. This way accurate visualizations of the actions of the robot and of the relevant aspects of its environment can be achieved.

1 Introduction

The Internet provides a unique opportunity to tele-operate and monitor mobile robots. Web-controlled mobile robots can give people all over the world the ability to become tele-present at distant places. Additionally, Internet-based tele-experimentation systems for mobile robots give distributed research groups located at distant places the ability to carry out joint experiments. This way, they can share expensive robot platforms and furthermore save travel expenses. Finally, the Internet can be used for on-line demonstrations with mobile robots, for example during the presentation of research results at conferences.

All these applications of Web interfaces for mobile robots require accurate visualization techniques. This includes a high level of detail as well as high update rates. Unfortunately, the Internet does only provide a restricted bandwidth. Therefore, video streams cannot be transferred at appropriate resolutions and with a frame rate required for smooth visualizations. Moreover, static monitoring cameras, which are frequently used on the Internet, have the disadvantage that they only provide a reduced field of view and that important details are occluded in certain perspectives.

In this paper we present state estimation techniques for a system which provides accurate and smooth real-time 3D visualizations of the movements of an autonomous mobile robot over the Internet. The system has a client/server architecture. The server is directly connected to the robot control system and transfers changes of the state of the robot and changes of the environment to all its clients which provide the 3D visualization at high frame rates. The system uses a 3D model of the robot and its environment. To cope with changes in the environment it uses probabilistic techniques to continuously estimate the states of different types of non-static objects such as people, doors, tables, etc. based on the information acquired by the robot's sensors. The changes of the environment are transfered to all the clients connected over the Internet which can instantly update the visualization.

2 Related Work

A variety of Web-based tele-operation interfaces for robots has been developed over the last few years. Three of the earlier systems are the Mercury Project, the "Telerobot on the Web", and the Tele-Garden [8, 9, 20] . These systems allow people to perform simple tasks with a robot arm via Web access. Since these manipulators operate in a well-known workspace without any unforeseen obstacles, all movement commands issued by a Web user can be carried out in a deterministic manner. Additionally, it suffices to provide still images from a camera mounted on the robot arm after a requested movement task has been completed. Our system, in contrast, is designed to visualize the actions of an autonomous mobile robot operating in a dynamic environment. In such environments the execution of a task generally depends on the current situation in the robot's environment.

Xavier [19] is a Web-operated autonomous mobile robot. It can be advised by Web users to move to an office and to tell a joke. The Web interface relies on client-pull and server-push techniques to provide visual feedback of the robot's movements; this includes images taken by the robot as well as a map indicating the robot's current position. However, Xavier's interface does not include any techniques to reflect changes of the environment.

The autonomous mobile robots Rhino and Minerva, which were deployed as interactive museum tour-guide robots in the „Deutsches Museum Bonn" in 1997 and in the Smithsonian Museum of American History in 1998, could also be operated over the Web [2, 18, 21]. In addition to image streams, recorded with on-board and off-board cameras, their interfaces offered Java applets for instant updates of information. Although the interfaces displayed information about the current actions of the robot and included applets providing smooth animations of the robot's trajectory in a map of the environment [18], the interfaces were restricted to 2D and changes in the environment were not displayed.

3D graphics visualizations for Internet-based robot control have already been suggested by Hirukawa et al. [10]. Their inter-

Auszug aus: Künstliche Intelligenz, Heft 4/00,
ISSN 0933-1875, arenDTaP Verlag, Bremen
www.kuenstliche-intelligenz.de
fon +49 421 34889-30 fax: +49 421 34889-31

face allows Web users to carry out manipulation tasks with a mobile robot, by controlling a 3D graphics simulation of the robot contained in the Web browser. In contrast to our system, these interfaces do not provide on-line visualizations of the robot's actions. Instead, they follow the tele-programming approach. Tasks are first tested off-line in the 3D simulation environment and are afterwards transmitted via the Internet for execution by the real robot.

In general, it is not sufficient to regard the environment as static and to display the robot within a static model of the environment. In practice, the environment of a mobile robot contains several non-static objects which influence the behavior of the robot. Among them are doors, chairs, and tables, which often change their position and can prevent the robot from taking a previously planned trajectory. To acquire and maintain a model of the environment is a major research area in mobile robotics. The most frequently used types of models are metric and topological maps. Topological models, as used in [13, 15], describe the environment at a coarse resolution. Because of the lack of necessary details, these types of models are only of limited use for visualizations. Metric maps, on the other hand, describe the environment at a finer level of detail. A popular approach are discrete occupancy grids proposed in [5, 14]. 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 such as doors, tables or chairs. In this paper we use a 3D 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 3D model so that the visualization can quickly be adapted to the current state of the environment.

3 State Estimation

To display the robot and its actions in the environment we use a 3D visualization system. This visualization system requires a 3D model of the robot's environment represented by a hierarchical scene graph, in which objects are described in the boundary representation. Figure 1 shows the 3D model of the environment our mobile robots generally operates in and in which the experiments described below were carried out. Given a corresponding description of the robot it can provide a visualization of the robot at arbitrary locations in the environment. Furthermore, the 3D model is used to compute simulated sensor measurements of the robot, which are needed by the state estimation components.

To estimate the state of dynamic objects, a mobile robot has to use its sensors. The information provided by the sensors, however, is inherently noisy so that state estimates relying on this information are uncertain. Therefore, it is not possible to exactly determine the state of an object. Instead, it is more appropriate to use probabilistic techniques and to maintain a probability density over the possible states of the objects. Within the Bayesian framework, such a density can be maintained over time, conditioned on the information gathered using the sensors. Obviously, for the rendering of the 3D world model, the actual states of the objects are required. Our method bases the decision about the current states of the objects on a maximum likelihood estimate.

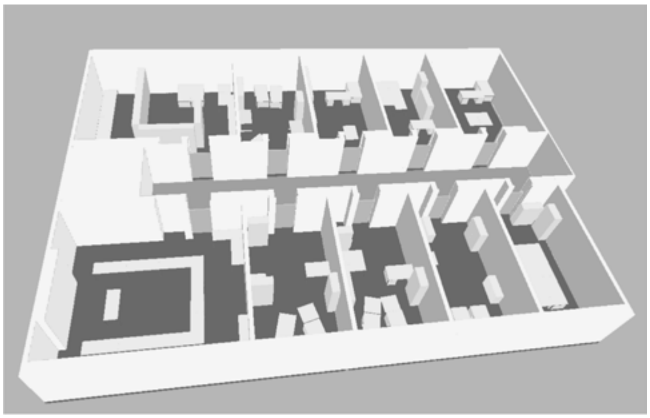

Figure 1: 3D Visualization (VRML) of the boundary representation of the robot's environment; The model contains walls and doors as well as furniture

In principle, we have to consider the complete state space of a dynamic world model so that we have to maintain the joint probability distribution over all possible states of the dynamic objects within the environment. Unfortunately, this approach is not tractable since the size of the joint state space grows exponentially in the number of objects. Therefore, we only consider marginal distributions and estimate the states of the dynamic objects independently.

Our approach distinguishes two different types of objects. The first class contains objects which are part of the world model and which are subject to infrequent state changes like doors, chairs, tables etc. The states of these objects are estimated using a template matching approach. The second class of objects contains objects which are continously moving through the robots environment like other robots or people walking by. Since such objects frequently enter and leave the scene, they are not permanent parts of the environmental model. Therefore, the state estimation method for these objects is based on the detection and tracking of dedicated features which indicate their presence. Furthermore, the independence assumption is often seriously violated for this class of objects, for example, if people walking by cross their path. In such a situation, we therefore have to solve the problem, which feature corresponds to which person. More generally, we have to assign features in the data to the objects being tracked. Our approach employs a probabilistic data association method for this purpose.

In the next sections, we introduce the general concepts underlying our state estimators before we describe the state estimation procedures.

3.1 Bayesian State Estimation

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

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

Here $p(s)$ is the prior density of the object's state. This prior is generally approximated based on the previous estimate. The whole process is known as Bayesian filtering. In the case of moving objects, we also use a motion model to update the density. The term $p(s\,|\,a,s')$ describes the probability that the object is in state s given it executed action a in state s'. The density is then updated according to the equation:

$$p(s|a) = \int p(s\,|\,a,s') \cdot p(s')\,ds'$$

Our system employs particle filters, whose basic idea is to propagate the object's state density over time using a sample-based representation of the density (see e.g. [4, 12, 17]). The two steps presented above are realized by the following two procedures.

1. Prediction
   In the prediction step, each sample is updated according to the model of the object's dynamics and the time t elapsed since the last estimate.
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 of the new observation given the sample state.

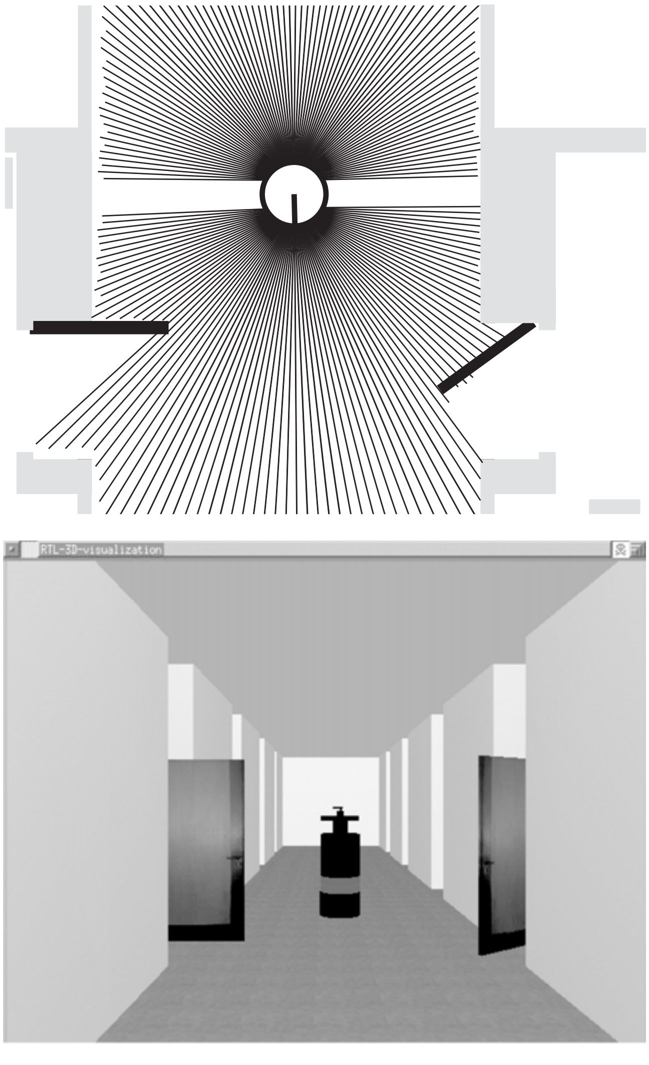

Figure 2: Rhino detecting the state of two doors with its laser range-finder (upper image). The estimated states of the doors are depicted in black. The resulting 3D-visualization is shown in the lower image.

3.2 Template-based State Estimation of Changing Objects

In order to determine the state of objects such as the opening angle of a door, the robot needs to know its own location l. To estimate $p(l)$, i.e. the probability that l is the robot's current position, we apply Monte Carlo localization, a method which is also based on the Bayesian framework described above, e. g. it maintains a probability density $p(l)$ over all the possible locations of the robot. To take the robot's uncertainty about its current position into account, we need to integrate over all locations during the Bayesian update:

$$p(s|o) = \int_l \frac{p(o\,|\,s,l) \cdot p(s) \cdot p(l)}{p(o|l)}\,dl$$

In this equation we assume that the state s of the object and the position l of the robot are independent. The denominator $p(o\,|\,l)$ can be be rewritten as

$$p(o|l) = \int_s p(o\,|\,s,l) \cdot p(s)\,ds$$

Thus, 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 the object and the location l of the robot. This likelihood can be derived by a template matching process which compares the current measurement to the measurements which are expected given the position of the robot and the state of the object. Our system uses a laser range-finder for state estimates and it computes the expected laser measurements using ray-tracing within the 3D world model (see Figure 2).

3.3 Tracking Moving Objects

To estimate the states of moving objects we apply a variant of a Baysian technique known as joint probabilistic data association filters (JPDAF). The standard JPDAF consists of a set of Kalman filters, one for each object being tracked, and an algorithm for associating features extracted from sensor data to the individual filters. The JPDA is a general approach to assign features detected in the data to the objects being tracked, where the number of objects being tracked is assumed to be known. The input to the JPDA is a set of features $\{o_1, ..., o_m\}$, which is extracted from the measurement o, and the output is an assignment of the features to the objects. The JPDA computes marginal probabilities β_{ji} of a feature o_j being originated from an object i by adding the probabilities of all possible assignments which assign feature o_j to the object with index i. Additionally probabilities β_{0i} are considered, which correspond to the events that no feature was detected for an object.

In our system the Kalman filters are replaced by particle filters. During particle filter update, all the features are considered for each object, but their contribution is weighted according to their association probability β_{ji}. The features used as input for the JPDA are characteristic local minima detected in the laser-range scan. Additionally it computes the difference in occupancy grid maps for consecutive scans obtained with the laser-range scanner as it is also done in [16] .

Auszug aus: Künstliche Intelligenz, Heft 4/00,
ISSN 0933-1875, arenDTaP Verlag, Bremen
www.kuenstliche-intelligenz.de
fon +49 421 34889-30 fax: +49 421 34889-31

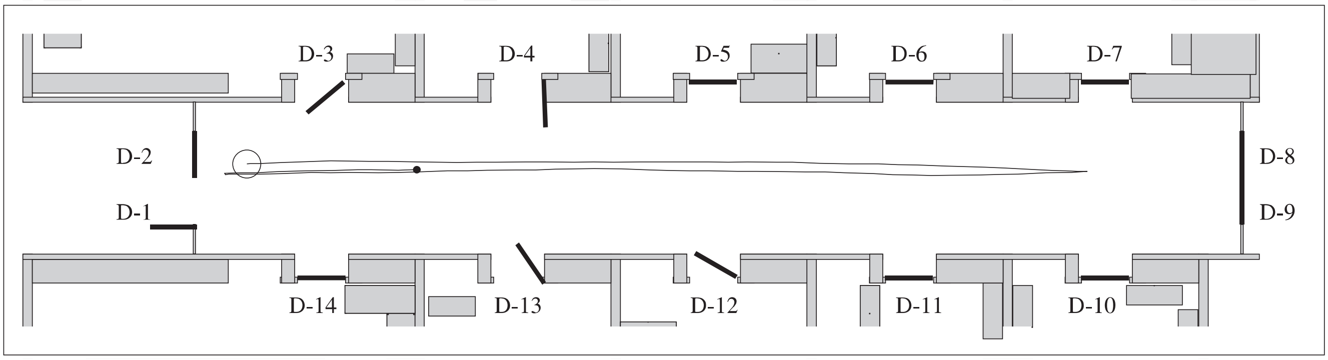

Figure 3: Trajectory of the robot; it started near D-2 and moved towards D-7 and back; also shown are the states of the doors during the experiment

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

Figure 4: Estimated states of the doors depicted in Figure 3; given are the means (avg. 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.

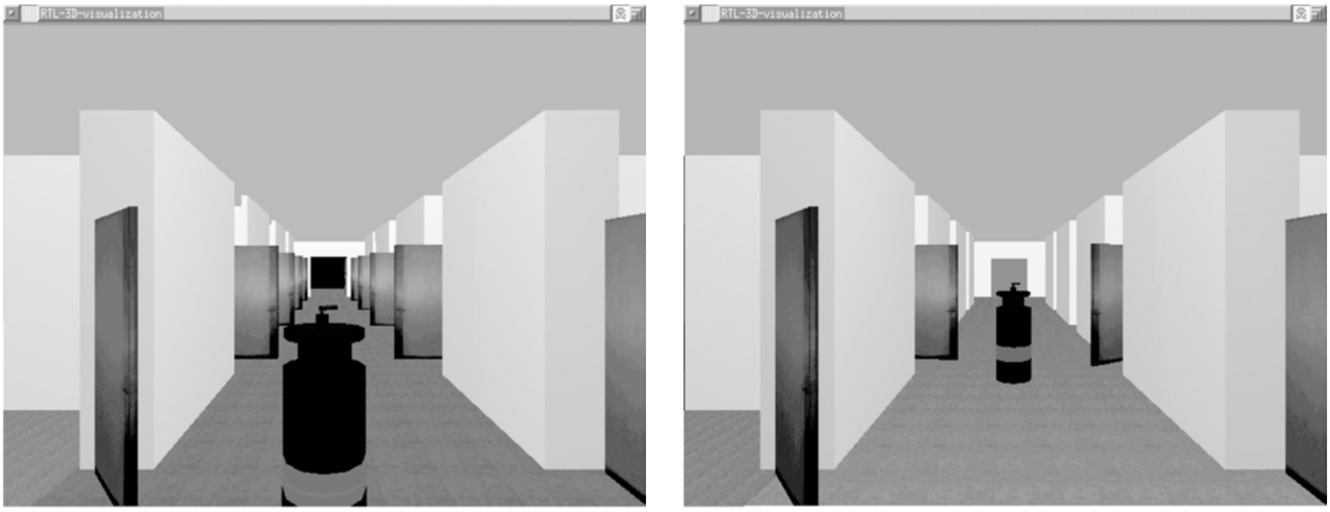

Figure 5: Visualization of the corridor at the beginning (left image) and at the end (right image) of the trajectory depicted in Figure 3.

4 Experimental Results

The system described in this paper has been implemented and tested in a typical office environment. The experiments described in this section were carried out with the mobile robot Rhino, a RWI[1] B21 robot. Rhino is equipped with two laser-range finders, each of which scans 180 degrees of the robot's surrounding at an angular resolution of 1 degree.

4.1 State Estimation for Changing Objects

To evaluate the state estimation technique and its advantages for a robust visualization of the robot's activities we performed several experiments in our office environment. In these experiments, the task of the robot was to estimate the state of the doors while the robot was traveling along the 26m long corridor of our department building.

To efficiently carry out the state estimation process, our current implementation uses a discrete set of possible states. For doors we typically use 10 states ranging from 0 to 90 degrees. That way, a uniform distribution is used as the prior density on all particle filter updates, e. g. the robot keeps no memory of the state of a door. To obtain more accurate estimates of the state of objects we integrate over a small set of different measurements. Figure 2 shows a typical situation in which the robot measures two doors. The resulting states of the doors are depicted in black. Figure 2 also shows the corresponding 3D visualization.

Figure 3 illustrates the trajectory of the robot as well as the states of the doors which were kept unchanged during the experiment. Figure 4 shows the results of the state estimation process for each door. As can be seen fromthese data our technique provides highly accurate results. In most of the cases the standard deviation is below 5 degrees.

[1] Real World Interface: http://www.rwii.com

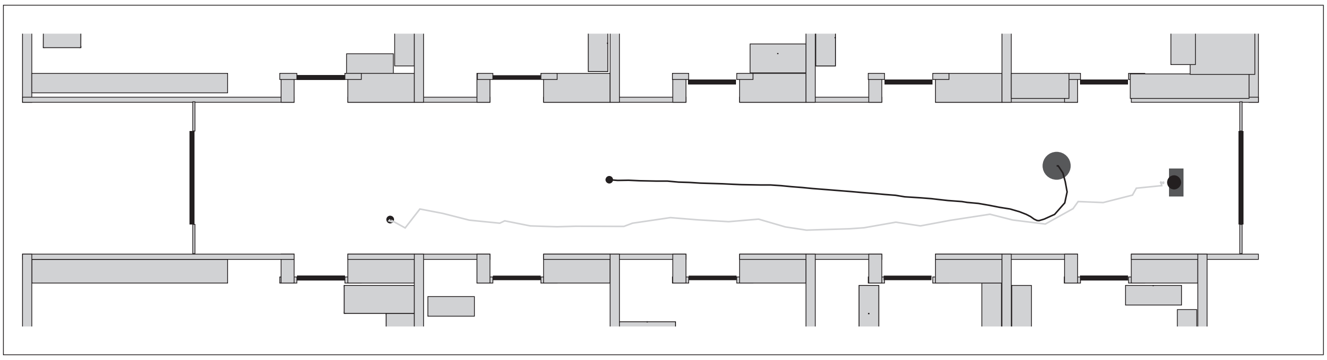

Figure 6: Trajectory of Rhino and estimated trajectory of a human passing by.

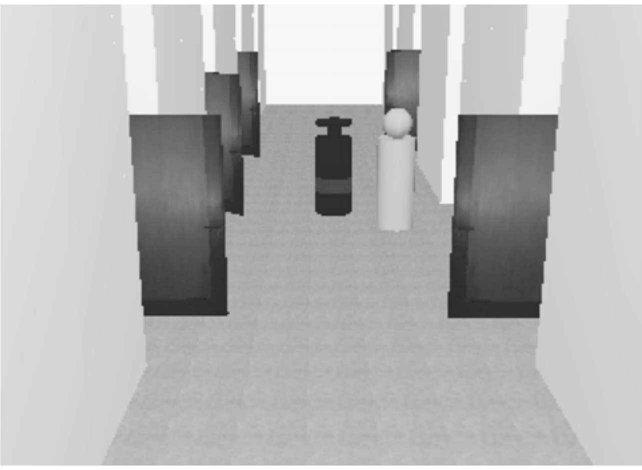

Figure 7: Left image: A 3D visualization of RHINO tracking a person; Right image: A photo of the situation at the same point in time

4.2 Tracking Moving Objects

To evaluate the performance of the tracking algorithm, we carried additional experiments in our office environment. Due to the limited angular resolution of 1 degree of Rhino's two laser range-finders, the current implementation limits the maximum range of the sensors to 7m. The resolution of the difference grid maps computed is 10 cm. Each particle filter uses a sample set consisting of 1000 samples. The implementation is highly efficient so that new laser-range scans can be integrated at a rate of 2 Hz.

In the experiments, the robot moved up and down the corridor of our department building, at a speed of 40 cm/s. While the robot was traveling, one and two people were moving within the robot's perceptual range of 7 m. We recorded the data obtained during these experiments and evaluated the tracking algorithm afterwards based on this data. Since no ground truth information is available for the trajectory of the humans, we can only give qualitative results about the tracking performance. Nevertheless, the results demonstrate that our technique is able to robustly keep track of moving objects over longer periods of time and even in situations in which features are missing.

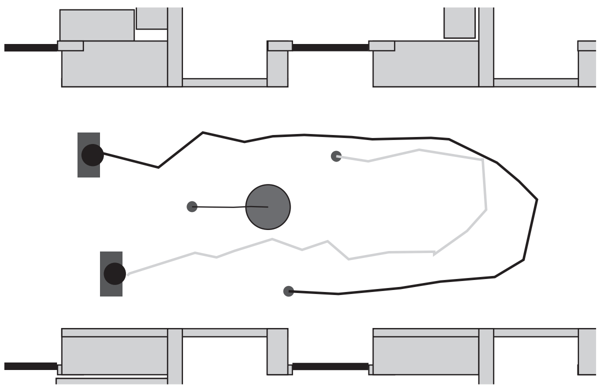

Figure 8: One person temporarily occluding a second one, in the back of the robot

Figure 6 shows a typical situation in which the robot is tracking a single person passing by the robot. Here the robot starts at the right end of the corridor, turns around, and moves to the left. At the same time the person passes by and walks in front of the robot. Whereas the human walked 17.6m at an estimated average speed of 50cm/s in this experiment, Rhino's trajectory was10.8m long and the robot had an average speed of 30cm/s. The longest continuous trajectory was approximately 102m, where the robot tracked the same person for 147 seconds. Figure 7 demonstrates that a 3D visualization is able to capture the experiment with sufficient detail and accuracy. The image on the right side of the figure shows a snapshot taken during a tracking experiment and the 3D image on the left displays the situation at the same point in time. The position and the parameters of the virtual camera were chosen to match the corresponding values of the real camera.

A particularly challenging situation is shown in Figure 8. Here, two people are approaching the robot. After passing it, they change sides and walk back. While they are changing sides, the person being located closer to the robot occludes the other one. The position of the robot depicted in this figure is the position at the time when the occlusion occurred. According to the estimates provided by the tracking algorithm, the two per-

Auszug aus: Künstliche Intelligenz, Heft 4/00,
ISSN 0933-1875, arenDTaP Verlag, Bremen
www.kuenstliche-intelligenz.de
fon +49 421 34889-30 fax: +49 421 34889-31

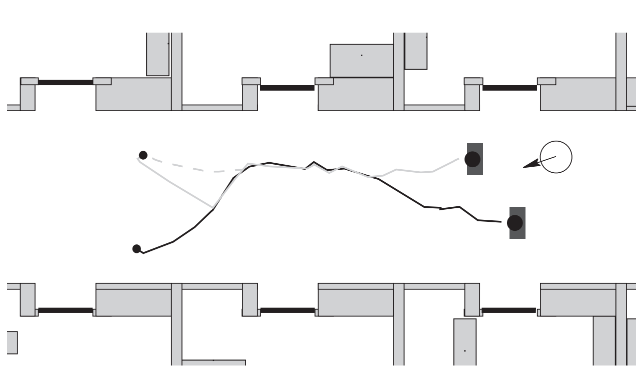

Figure 9:Two persons walking close to each other

son were moving at a speed of 60 cm/s and 80 cm/s, respectively. This experiment was repeated several times and the system was always able to keep track of the two persons.

Another difficult situation is depicted in Figure 9. Two people are joining in front of the robot and walk next to each other for a few seconds before the trajectories split-up again. The problem here is that the state densities of the two persons are nearly identical. They walk at the same speed, into the same direction, and their positions only have a small displacement. Nevertheless, our technique is able to keep track of both persons as Figure 9 shows. However, the point in time, at which the to persons split-up is detected too late. The correct trajectory of one person is illustrated by the dashed line. The trajectory estimated by the robot is shown in light grey.

5 Conclusions

In this paper we presented an approach for the robust 3D visualization of the actions of a mobile robot over the Internet. Instead of image streams our technique uses a 3D model of the environment and visualizes the actions of the robot and the current state of the environment by 3D rendering. According to that, only the current state of the robot and of dynamic objects within the environment have to be transferred over the Internet which results in a serious data reduction. Our approach applies a probabilistic state estimation technique to continuously determine the current states of dynamic objects while the robot is moving. The system has been tested extensively in a typical office environment. The results show that our state estimation procedure can robustly and efficiently estimate the state of different dynamic objects in the environment of the robot. As a result, we obtain accurate visualizations of the robot and its environment.

The system and techniques described in this paper can contribute to a variety of applications. For example, distributed research groups carrying out joint experiments, can use the 3D visualization to monitor the actions of the robot. The techniques can also be used for the documentation of experiments and to illustrate the behavior of a robot using a certain control strategy. Finally, we are convinced that such techniques will play a major role in future Web-based tele-presence systems. In the TOURBOT project, for example, a mobile robot is used to allow users the inspection of objects in exhibitions from arbitrary viewpoints over the Web. In this context, 3D visualization techniques will be used to visualize the current location of the robot.

There are several aspects for future research. For example, using 3D visualizations has the advantage that a user can watch the robot from arbitrary perspectives by choosing view-points for the virtual camera. However, finding appropriate view-points is not at all an easy task, especially for non-expert users. The system therefore requires techniques for selecting view-points automatically. One approach towards this direction, is described in [11]. A second important topic regards the acquisition of the 3D model. The model used by our current system has been hand-crafted in a time-consuming process of several days. In this context it would be worthwhile to have techniques for the automatic acquisition of 3D maps by the robot itself. First steps towards this direction have been presented in [22]. However, these techniques are sub-symbolic and do not provide means for detecting objects like doors, chairs etc. which are used by the system presented in this paper.

Acknowledgments

This project has been supported by the North-Rhine-Westphalia Ministry for Schools, Education, Science and Research (MSWWF-NRW, Germany), under contract number IVA3-10703596 and by the IST Programme of Commission of the European Communities under contract number IST-1999-12643. The authors furthermore would like to thank Dieter Fox, Michael Beetz, and Sebastian Thrun as well as the members of the research lab for autonomous mobile systems at the University of Bonn for fruitful discussions.

References

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Mathematics in Science and Engineering. Academic Press, 1988.

[2] W. Burgard, A.B. Cremers, D. Fox, D. H¨ ahnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 1999.

[3] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.

[4] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical Report CUED/F-INFENG/TR.310, Department of Engineering, University of Cambridge, 1998.

[5] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Trans-actions on Robotics and Automation*, 3(3):249–265, 1987.

[6] Panos Trahanias et al. TOURBOT– interactive museum tele-presence through robotic avatars. In *Proc. of the 9th International World Wide Web Conference*, May 2000. to appear http://www.ics.forth.gr/tourbot/.

[7] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1999.

[8] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Desktop tele-operation via the world wide web. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1995.

[9] K.Goldberg, J.Santarromana, G. Bekey, S.Gentner, R.Morris, J. Wiegley, and E. Berger. The telegarden. In *Proc. of ACM SIGGRAPH*, 1995.

[10] H. Hirukawa, T. Matsui, and H. Onda. Prototypes of teleoperation systems via a standard protocol with a standard human interface. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1997.

[11] A. Hopp, D. Schulz, W. Burgard, A.B. Cremers, and D. Fellner. Virtual reality visualization of distributed tele-experiments. In *Proc. of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 1998.

[12] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. of the European Conference of Computer Vision*, 1996.

[13] D. Kortenkamp and T. Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proc. of the Twelfth National Conference on Artificial Intelligence*, pages 979–984, 1994.
[14] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, Summer 1988.
[15] D. Pierce and B. Kuipers. Learning to explore and build maps. In *Proc. of the Twelfth National Conference on Artificial Intelligence*. AAAI Press, 1994.
[16] E. Prassler, J. Scholz, M. Strobel, and P. Fiorini. MAid: A robotic wheelchair operating in public environments. In *Proc. of International Workshop on Sensor-Based Intelligent Robots*, LNCS. Springer Verlag, 1998.
[17] D.B. Rubin. Using the SIR algorithm to simulate posterior distributions. In M.H. Bernardo, K.M. an DeGroot, D.V. Lindley, and A.F.M. Smith, editors, *Bayesian Statistics 3*. Oxford University Press, Oxford, UK, 1988.
[18] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B. Cremers. Web interfaces for mobile robots in public places. *IEEE Robotics & Automation Magazine*, 7(1):48–56, 2000.
[19] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan. A layered architecture for office delivery robots. In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, February 1997.
[20] K. Taylor and J. Trevelyan. A telerobot on the World Wide Web. In Proceedings of the 1995 National Conference of the Australian Robot *Association*, 1995.
[21] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second generation mobile tour-guide robot. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1999.
[22] 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.

Contact
Dirk Schulz
Department of Computer Science, University of Bonn,
D-53117 Bonn, Germany
Email: schulz@uran.informatik.uni-bonn.de

Wolfram Burgard
Department of Computer Science,
University of Freiburg, D-79110 Freiburg, Germany
Email: burgard@informatik.uni-freiburg.de

Prof. Dr. Armin B. Cremers
Institut für Informatik III
Universität Bonn, Römerstr. 164, D-53117 Bonn
Email: abc@cs.uni-bonn.de

Dirk Schulz is a Ph.D. student at the Department of Computer Science III at the Universtity 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.

Armin B. Cremers is a full professor of computer science and chair of the Department of Computer Science III at the University of Bonn. Prior to this position he was a full professor at the University of Dortmund and an assistant professor at the University of Southern California. His scientific research is in the areas of database and information systems, autonomous mobile robots and artificial intelligence.