In *Proc. of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, 1998*

A Hybrid Collision Avoidance Method For Mobile Robots

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

[†]Dept. of Computer Science III, University of Bonn, D-53117 Bonn

[‡]Dept. of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

Email: {fox,wolfram}@cs.uni-bonn.de, thrun@cs.cmu.edu

Abstract— **This paper proposes a hybrid approach to the problem of collision avoidance for indoor mobile robots. The μDWA (short for: model-based dynamic window approach) integrates sensor data from various sensors with information extracted from a map of the environment, to generate collision-free motion. A novel integration rule ensures that with high likelihood, the robot avoids collisions with obstacles not detectable with its sensors, even if it is uncertain about its position. The approach was recently implemented and tested extensively as part of an installation, in which a mobile robot gave interactive tours to visitors of the "Deutsches Museum Bonn." Here our approach was essential for the success of the entire mission, because a large number of ill-shaped obstacles prohibited the use of purely sensor-based methods for collision avoidance.**

I. Introduction

In order to operate safely in populated environments, many successful mobile robot systems rely on fast, sensor-based collision avoidance modules to control the robot (see e.g. [12, 2, 7, 16, 11, 9]). The predominant paradigm of these approaches is strictly sensor-based: Sensor readings are continuously analyzed to determine collision-free motion.

Unfortunately, the sensor-based paradigm has important limitations. If the environment is complex, it might be difficult to equip a robot with a sensor suite capable of detecting arbitrary obstacles. For example, if the environment possesses large obstacles made of glass (such as in our testing environment, see below), light-based sensors will not be able to detect them and even sound-based sensors such as sonars usually have severe problems due to specular reflections, which often occur at smooth surfaces such as glass. The severity of the problem increases with the speed of the robot, as obstacles have to be detected early enough to allow the robot to decelerate safely.

In a recent attempt to move away from office-type environments into more difficult ones, we found the problem of undetectable obstacles to be a major obstacle in the way of successful mobile robot navigation. In particular, we recently installed our mobile robot RHINO [3, 19] in the *Deutsches Museum Bonn* (German Museum Bonn), where it served the function of an interactive robotic tour-guide. The robot's task was to engage visitors and guide them through the exhibition, providing verbal explanations for the various exhibits (see Figure 1). What made this task specifically challenging was the nature of the environment. While RHINO is equipped with five different sensor systems (see Figure 2), various obstacles were virtually undetectable for the robot, such as: glass cages, put up to protect exhibits (the label $o1$ marks such a barely visible cage in Figure 2), metal bars at various heights (see label $o2$ in Figure 2), small socles or metal plates on which exhibits were placed (e.g., $o3$ in Figure 1), just to name some. For the museum tour-guide application to be successful, the robot had to move at walking speed. Avoiding collisions was of uttermost importance due to the nature of the "obstacles" in a museum. The reader may notice that similar conditions are expected to be found in private homes and various other anticipated task environments for future service robots.

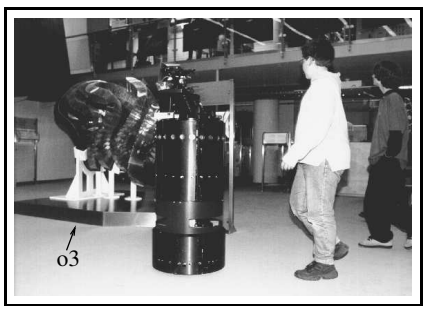

Fig. 1. RHINO, as it gives a tour through the museum. The label $o3$ points to a hard-to-detect obstacle.

This paper proposes a hybrid approach to collision avoidance, called μDWA (short for *model-based dynamic window approach*), which integrates our strictly sensor-based dynamic window approach [7], with a map of the environment. The location of the map (and hence the obstacles) relative to the robot is estimated using a metric version of the recently proposed *Markov-localization* algorithm, an algorithm that has been shown to be extremely robust and reliable for position es-

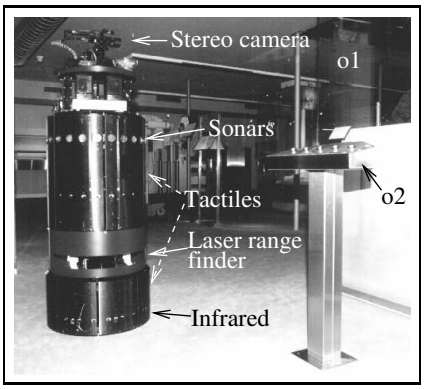

Fig. 2. Sensors of the robot RHINO. The label $o1$ highlights an almost invisible glass surface, and the label $o2$ shows a metal console which is just below the robot's sonar sensors.

timation in populated environments [15, 17, 10, 5, 8]. Markov-localization is a probabilistic algorithm that outputs a probability density function over all possible positions, rather than a single position. A key feature of Markov localization is that it can represent ambiguities and handle uncertainty in a mathematically consistent and elegant way.

μDWA uses a conservative, probabilistic integration rule to provide maximum safety in situations where the robot is uncertain about its actual position. If the robot knows its position with absolute certainty, μDWA degrades to the obvious extension of DWA using a map; however, if the robot is unable to disambiguate its position (which typically occurs in crowded environments), μDWA is guaranteed to avoid collisions with high (e.g. 99%) probability. μDWA has been successfully tested for more than 18.5 km in a crowded museum, where it has been essential for RHINO's safety, as elaborated further in the experimental results section of this paper.

The remainder of this paper reviews the DWA approach to collision avoidance, as previously used successfully for various applications within office environments [7, 19]. It then describes our implementation of Markov localization, which differs from previous implementations in that (1) it estimates the robot's location in fine-grained metric coordinates and not in coarse-grained topological entities—which is crucial for the approach described in this paper–, and (2) it does not rely on specific geometric properties of the environment (previous approaches required entities such as hallways, openings, and so on, most of which did not exist in the Deutsches Museum). Finally, it describes the μDWA approach for avoiding collisions by generating *virtual* sensor readings using a geometric map of the environment. The paper concludes with a description of some experimental results and a discussion of the implications and limitations of this research.

II. The Dynamic Window Approach

The *Dynamic Window Approach* (in short: DWA) has recently been proposed for collision avoidance for high-speed (up to .95 cm/sec) indoor navigation [7, 6, 16]. It differs from the majority of work in the field in that it does not consider the robot a kinematic entity that can move in arbitrary directions at any point in time. Instead, DWA models robots as dynamic objects, paying specific attention to the dynamic constraints imposed by the inertia of a fast moving system. Here we will only review the key ideas of the algorithm; see [7] for more details and various experimental results.

The key idea in DWA is to choose control in the *velocity space* of the robot. In DWA, the velocity space of synchro-drive robots is parameterized by the *translational* and *rotational* velocity. As shown in [7], robots with fixed velocity (no torque) always travel on a circular trajectory (whose diameter is determined by the ratio of translational and rotational velocity). Motor current (torque) change the velocity of the robot and, as a consequence, its motion direction.

In regular time intervals (e.g., every .25 seconds), DWA chooses velocities so as to best obey various hard and soft constraints:

Hard constraints are vital for a robot's safety and are imposed by torque limits. For example, the maximum torque induces a maximum change of velocity, which severely limits the space of possible control (e.g., a fast moving robot cannot take a 90 degree turn). Hard constraints are also imposed if a velocity would inevitably lead to a collision with an obstacle. Hard constraints rule out certain controls from further consideration. Notice that hard constraints do not specify preferences among the different control options; neither do they take into account the robot's task.

Soft constraints express *preferences* for both the motion direction and the velocity of the robot. DWA utilizes three different soft constraints, which measure (1) progress towards the goal, (2) forward velocity, and (3) forward clearance. If combined in the right ratio, these criteria lead to goal-directed behavior with freedom to graciously move around obstacles.

In previous experiments [7], DWA was found to yield safe robot navigation in various indoor office environments, at speeds of up to 95 cm/sec, and using robots equipped with sonar sensors, laser range finders, or both. In fact, the DWA approach has been adopted by a leading mobile robot manufacturer, Real World Interface, Inc., as their sole collision avoidance package for their B14/B21 robots. It is now in use at more than 15 academic institutions. As noticed above, DWA is purely sensor-based. μDWA extends DWA by a map-based component, as described in turn.

III. Metric Markov Localization

The central problem in integrating map-based information into collision avoidance is the problem of *localization*, which is

the problem of determining a robot's position relative to its map. While many approaches for mobile robot localization have been proposed in the literature (see [1] for a comprehensive overview), the majority of approaches is too brittle and/or depends on specific assumptions/modifications of the robot's environment. Recently, Markov localization has been proposed and implemented with considerable success by several groups [15, 17, 10, 5, 8]. Markov localization uses probabilistic data structures to estimate a robot's position, enabling it to deal with perceptual limitations, uncertainty and ambiguity in a principal and mathematically elegant way. For example, if past sensor readings are insufficient to uniquely determine the robot's location, Markov localization assigns high likelihood to multiple locations. As a result, this family of approaches exhibit improved level of robustness, as demonstrated in [5, 19, 13].

Markov localization differs from most traditional approaches in that the robot does not represent its internal belief by just a single position. Instead, it represents *all* possible positions, where each position is weighted by a likelihood factor. To see, let l denote a location in x-y-θ space where x and y are Cartesian coordinates and θ is the robot's orientation (all relative to the map). Markov localization maintains a probability density function, denoted by P, that models the robot's belief of being at the different positions in the environment. Initially, the position of the robot relative to its map might be entirely unknown. In such cases P is initialized uniformly. As the robot senses, it updates P using the following equations, which are easily derived using Bayes rule or Markov chain theory (see [5] for a derivation):

$$P(l) \quad \longleftarrow \quad \frac{P(s \mid l)\ P(l)}{P(s)} \tag{1}$$

where $P(s)$ is a normalizer that ensures that the probabilities $P(l)$ over all l sum up to 1. Here s denotes a sensor measurement and $P(s \mid l)$ is the probability of measuring s at location l. In μDWA, the sensors are assumed to measure proximity and s are proximity measurements (obtained from laser range finders and/or sonar sensors). $P(s \mid l)$ is obtained using the map and a simplistic sensor model, which is described in more detail in [4].

When the robot moves, P is convolved using a probabilistic model of robot motion:

$$P(l) \quad \longleftarrow \quad \sum_{l'} P(l \mid u, l')\ P(l') \tag{2}$$

where $P(l \mid u, l')$ denotes the probability that the robot is at l upon executing control u at position l'. In μDWA, $P(l \mid u, l')$ is implemented by a bounded-Gaussian distribution centered at the geometrically expected position. These two equations are sufficient to refine a robot's belief upon sensing and moving; they are at the core of the various implementations of Markov localization.

In most existing implementations, P is represented discretely, where each location corresponds to a node in a pre-supplied coarse-grained, *topological* map of the environment [15, 17, 10, 8]. μDWA employs a *geometric* variant of Markov localization. More specifically, P is represented by a fine-grained, regularly spaced grid, where the spatial resolution is usually between 10 and 15 cm and the angular resolution is usually 1 or 2 degrees. The advantage of such a high resolution is obvious: To avoid collisions reliably, the robot needs highly accurate position information.

At first glance, one might be inclined to think that a disadvantage of the geometric approach lies in its computational complexity: An environment of size 30×30 m^2 with a spatial resolution of 15 cm and an angular resolution of 2° possesses approximately $7.2 \cdot 10^6$ discrete entities. While this is generally true, a variety of additional techniques has been developed in our previous work to update such large tables in real-time, while the robot is in motion. Among these techniques, two are most essential: (1) The various conditional densities are stored as fast look-up tables whose access is extremely efficient, and (2) instead of computing probabilities for *all* locations, the robot selectively updates only the most likely ones. These modifications sped up the basic algorithm by several orders of magnitude, making it possible to estimate the robot's position in real-time, while the robot is in motion. In extensive experimental tests, we did not observe evidence that these modifications impact the robot's behavior in any noticeable way. Basically, most of the time the likelihood of almost all positions is so close to zero, that not updating them has almost no effect on the resulting belief; yet it reduces the computational complexity by orders of magnitude.

Figure 3 shows an outline of the exhibition in the *Deutsches Museum Bonn* where RHINO served as a tour-guide. The size of the fraction of the museum where RHINO gave tours was approximately 20×20 m^2. The figure also shows the position probabilities P during global localization (darker positions are more likely). Several local maxima in the distribution show that the position of the robot is not yet uniquely determined. During localization, the certainty of the position estimation increases and the density typically concentrates on the real position of the robot (see Figure 4). As noticed above, all computation is carried out in real-time, while the robot moves. Often, each sensor scan is processed in less than .1 seconds, using a 200MHz Intel PentiumPro. Localizing the robot from scratch (without telling it where it is) requires less than two minutes.

IV. Model-Based DWA

The key idea of the *model-based dynamic window approach* (μDWA) is the integration of real and "virtual" sensor data, derived from a map of the environment. To translate the map into local, robot coordinates, the robot must know where it is. If the robot always knew its position with absolute certainty, integrating maps into sensor-based collision avoidance would be

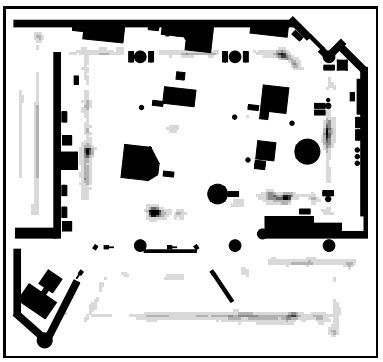

Fig. 3. Map of the museum (black) along with position probabilities (grey). See text.

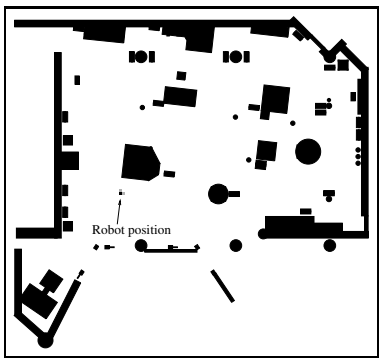

Fig. 4. Position probability when the robot is highly certain.

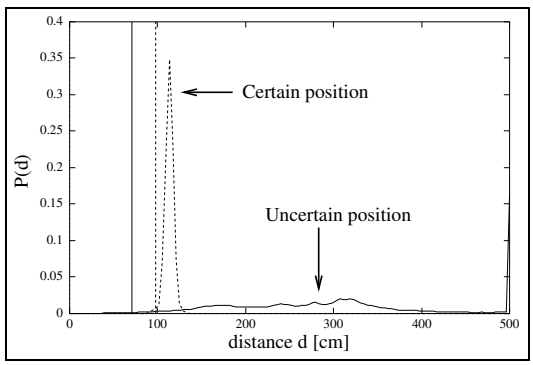

Fig. 5. Two examples of $P(d)$ as a function of d. See text.

straightforward. Markov localization, however, does not produce a single estimate; instead, it provides a probability density function over all possible robot positions that reflect the robot's belief of actually being there. If the robot is not quite certain about its position, the probability density function might be multi-modal with peaks at the most plausible locations. This raises the important question as to how to best synthesize the "virtual" sensor data. On the one hand, one wants to ensure safe operation even with high probability. On the other hand, one does not want to restrict the robot's freedom too much, even when it is uncertain as to where in the world it is.

In the following, we introduce a probabilistic representation of a *perfect* virtual sensor "mounted" at angle α relative to the robot's coordinate system. Let $d_\alpha(l)$ be the distance to the nearest obstacle when the robot's position (in x-y-θ space) is l. $d_\alpha(l)$ can be computed using a map.[1] Furthermore, let X_α denote a random variable that models a measurement of such a virtual sensor. Obviously, if l is known the probability $P(X_\alpha = d \mid l)$ that this sensor returns a value $d \geq 0$ is 1 if $d = d_\alpha(l)$ and 0 otherwise.

$P(X_\alpha = d \mid l)$ assumes knowledge of the robot's position. Suppose the position l of the robot is unknown; instead, one is given a belief $P(l)$ about its current position. Then, the probability that the sensor returns a value d and the probability that the sensor returns a value larger than d is given by Equations (3) and (4), respectively.

$$P(X_\alpha = d) \quad = \quad \sum_l P(X_\alpha = d \mid l)\, P(l) \tag{3}$$

$$P(X_\alpha > d) \quad = \quad \sum_{d_i > d} P(X_\alpha = d_i) \tag{4}$$

Figure 5 depicts the density of the measurement X_α in the two situations shown in Figures 3 and 4: one, in which the robot does not know its position well, and one where it is fairly certain about its position. As can easily be seen, when the robot is uncertain about its position, X_α is spread over many different measurements (solid line). If the robot knows its position well, the distribution of X_α is centered around a single distance (dashed line). In both cases, however, the robot assigns non-zero likelihood to extremely short measurements, since Markov localization never excludes a position with absolute certainty.

μDWA selects the "virtual" measurements using a conservative rule: *The virtual measurement of a sensor is the largest distance d^*, such that with probability .99 a distance larger than d^* is measured*:

$$d^* \quad = \quad \max\{d \mid P(X_\alpha > d) \geq .99\} \tag{5}$$

The vertical lines in Figure 5 illustrate this value for the corresponding density (imagine the robot being on the left side of each plot). By conservatively picking a sensor value that, with high probability, is shorter than the proximity of the obstacle, the robot is likely to avoid a collision even in the face of uncertainty. Notice that our approach provides maximum freedom under the constraints of 1% error probability.[2]

For collision avoidance, the virtual sensor has to be fired frequently (e.g. every 50 cm of robot motion) into all directions. In our implementation of μDWA we have modified the basic code to fulfill this task in real-time. The most important modification concerns the computation of the density of the measurement X_α: instead of integrating over all locations l in Eq. (3), only a subset of the all possible locations is considered, including only cells with probability above a threshold. The threshold is set such that these cells in most cases represent more than 99% of the position probabilities. Our simplification is somewhat justified by the observation that in practice, P is

[1] In fact, in our implementation $d_\alpha(l)$ is computed in advance for all possible l and α and stored in a look-up table, which maximizes run-time efficiency. See [4] for a more detailed discussion of efficient retrieval.

[2] Alternative schemes, such as picking the minimum distance among those locations l whose likelihood is above a certain threshold are *not* guaranteed to yield the same probabilistic bound in the likelihood of failure.

usually quickly centered on a small number of hypotheses and approximately zero anywhere. In the worst case, one can show that this modification yields an additional 1% error probability, lowering the probabilistic safety bound to 98%.

V. Experimental Results

A. Museum Tour-Guide Project

μDWA was tested extensively in a recent installation in the Deutsches Museum Bonn, where the mobile robot RHINO was deployed as an interactive tour-guide robot. Here the robot's task was to engage visitors and to guide them through the museum, providing verbal explanations for the various exhibits. Safe navigation was of uttermost importance, since RHINO is strong and heavy enough to severely violate children, and since some of the exhibits were unique or extremely expensive.

Several factors contributed to the difficulty of the problem:

1. **Invisible obstacles.** As noticed in the introduction to his paper, various obstacles were basically "invisible" to the robot, despite the fact that our robot applied four different sensor systems for obstacle detection (c.f. Figure 2).

2. **Speed requirements.** To be "interesting" to people, the robot had to navigate at least at walking speed.

3. **Dynamic obstacles.** Large crowds often blocked much of the free space, and they often challenged the robot in various ways. Operating on a pre-planned, static path was not feasible. Instead, the robot had to continuously assess the situation and plan its motion accordingly.

4. **Sensor blockage.** The large number of people also made accurate localization a difficult and challenging problem, since they often blocked RHINO's sensors for extended durations of time (see Figure 6).

5. **Lack of features.** The problem of localization was particularly difficult in the center portion of the environment, a large open space that mostly lacked features necessary to determine the robot's position.

Avoiding collisions was clearly more difficult than in any of the various office environments in which our software was previously developed and tested.

During a total of 47 hours within six days of robot navigation, μDWA proved to be highly reliable, and was clearly essential for the success of the entire system. Figure 6 shows a map of the museum. Here grayly shaded areas indicate obstacles that can only hardly (or even not at all) be perceived using the robot's sensors. Shown also is a path that the robot took. This path is approximately 1.6 km long, and captures 4.5 hours of robot motion. When the robot was not explaining an exhibit, it moved at an average speed of 36.6 cm/sec. In crowded situations, the average velocity was often lower; however, at times the robot moved at speeds of 70 cm/sec or higher for extended durations of time. The results of the entire project are summarized in Table 1: In the six days of the museum tour-guide project RHINO traveled more than 18.6 km. Whenever possible, it chose its maximum speed of 80 cm/sec. Although the robot's path was frequently blocked by visitors, RHINO kept an average speed of 36.6 cm/sec when traveling from one exhibit to another. More than 2,000 real visitors and over 600 "virtual" Web-based visitors[3] were guided by RHINO, some of whom followed the robot for more than an hour. RHINO fulfilled 2,400 tour requests by real and virtual visitors of the museum. Only six requests were not fulfilled (mostly due to scheduled battery changes at the time of the request), which lead to an overall success-rate of 99.75%.

This project demonstrates the reliability of μDWA. During the entire project, the robot never collided with any of the visitors. We counted a total of six collisions with exhibits in the museum, all of which were minor and neither of them caused any damage. Out of those six collisions, only one was directly related to μDWA: Here an "invisible" obstacle was approximately 20 cm closer than μDWA had determined, causing the robot to touch the metal platform of one exhibit ($o3$ in Figure 1). This incident was preceded by a failure of a major sensor system which introduced error into the localization (the duration of this failure is not known to us, but we actually observed the failure before the collision occurred). Three other collisions were caused by hardware problems (such as low battery power). The remaining two collisions were caused by flaws in the hand-crafted map, which initially lacked some essential obstacle information.

B. The Role of Probabilistic Integration

A key aspect of μDWA is its ability to generate virtual sensor readings even if the robot does not know where it is (c.f., Equations (4) and (5)). To illustrate the importance of considering the entire distribution P instead of just a single estimate, we empirically compared μDWA to an approach which only considers the *most likely* robot position ($\operatorname{argmax}_l P(l)$) to generate virtual sensor readings. This approach can be thought of as the logical counterpart of μDWA if the localization component is not probabilistic and just maintains a single estimate.

Our experiments indicate that μDWA's integration is safer when the robot is not certain about its location. The upper part of Figure 9 shows a map of one of our testing environments, a mostly symmetric office environment in our university building. The situation shown there is one where the robot has not been able to uniquely determine its location.

While the robot is truly at the location labeled a, it assigns slightly higher probability to the location labeled b. Such situations often occur in symmetric environments, specifically if the robot is not told its initial position (as was the case in this

[3] `http://www.cs.uni-bonn.de/~rhino/tourguide/`

Fig. 6. A typical situation: Rhino seeks its way through the crowd.

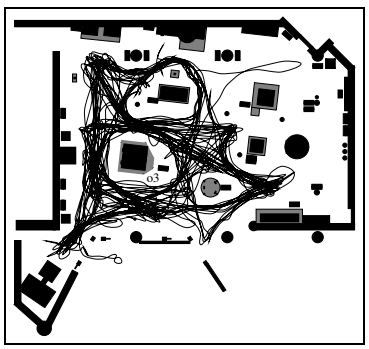

Fig. 7. Path of the robot during a single 4.5 hour run (1.6 km).

Hours of operation	47
Number of visitors	> 2000
Kilometers traveled	18.6
Maximal speed of travel	> 80 cm/sec
Average speed during motion	36.6
Number of collisions	6
Number of requests	2400
Percentage of completed requests	99.75

Tab. 1. Some key figures from the museum tour-guide project.

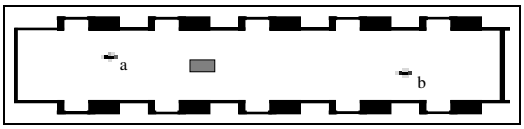

Fig. 9. Ambiguous situation in a corridor.

particular experiment). Here the advantage of μDWA is obvious: While the maximum likelihood approach considers exclusively location b when generating the virtual sensor readings, μDWA takes both potential locations into account, thus picking the most conservative virtual sensor reading. To see, consider the Figure 10. Here the dashed lines shows $P(d)$ when averaged over all locations (as in μDWA), whereas the solid line shows $P(d)$ determined on the most likely estimate only. As a result, the maximum likelihood approach will falsely generate long reading, whereas μDWA will generate a reading that prevents the robot from colliding. For μDWA to err, the robot has to assign less than 2% probability to the correct location—something that we observed only once, throughout all our experiments.

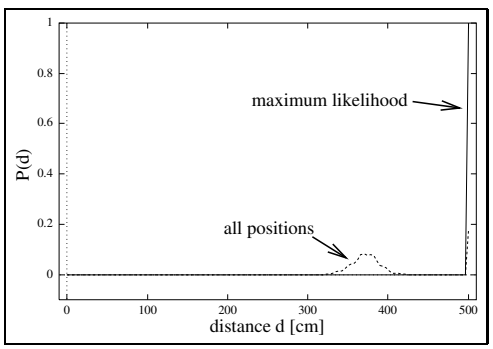

Fig. 10. Distance probabilities in an ambiguous situation.

VI. Discussion

This paper described a hybrid approach to collision avoidance, called μDWA, which integrates both sensor data and data from a previously supplied map into collision avoidance. This approach combines the best of both worlds: it reacts adequately to unexpected obstacles (such as humans), but it also avoids collisions with undetectable obstacles whose locations are known. It has been tested extensively in various densely populated environments (including a museum), in which a large number of obstacles (exhibits) were impossible to detect with the robot's sensors.

The work presented here has significant impact on future low-cost robot applications. The ability to integrate map-based information into collision avoidance, even if the robot is not certain about its actual location, reduces some of the burden to equip robots with potentially high-cost sensors. For example, in (ongoing) experiments using data recorded in the museum, we have found strong evidence that sonar sensors alone would have been sufficient for localization, and thus for collision avoidance. Such a finding suggests the feasibility of installing robots that only use sonar sensors (instead of the much more expensive laser sensors) in environments as complex and unstructured as this specific museum. Hybrid approaches to collision avoidance, which react to sensor readings but also consider models of the environment, have not received much attention in the literature. However, we believe that many environments require such hybrid approaches, of which the Deutsches Museum is certainly one.

The current approach also suffers limitations, which mainly arise from the need of an accurate metric map. In the particular experiments reported here, the map was constructed manually, using measuring tape. In most robot applications such an approach is justifiable by the fact that the installation costs (i.e., acquiring a map) are small compared to the day-to-day operational costs. However, in some environments, such as private homes, the need for a metric map might make it difficult to apply the method described here. Recent research on

map acquisition [19, 18] provides a way to acquire the map autonomously. While the real-time requirements for reactive collision avoidance prohibit the use of time-consuming sensor-interpretation techniques such as complex visual scene interpretation, such techniques can be applied to improve the model of the environment. Of course, methods for map acquisition model only those obstacles that can be detected by the robot's sensors. It appears to be feasible, however, to "label" either undetectable obstacles or even forbidden areas by driving the robot (manually) along the boundary of its legal operational space. With such an approach, μDWA should be applicable even in many domains were up-front map information is unavailable. The feasibility of this approach is subject to future research.

REFERENCES

[1] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[2] J. Borenstein and Y. Koren. The vector field histogram - fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3), 1991.

[3] J. Buhmann, W. Burgard, A.B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot Rhino. *AI Magazine*, 16(2), Summer 1995.

[4] W. Burgard, D. Fox, and D. Hennig. Fast grid-based position tracking for mobile robots. In *Proc. of the 21th German Conference on Artificial Intelligence (KI 97), Germany*. Springer Verlag, 1997.

[5] W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the Fourteenth National Conference on Artificial Intelligence*, 1996.

[6] D. Fox, W. Burgard, and S. Thrun. Controlling synchro-drive robots with the dynamic window approach to collision avoidance. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[7] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), March 1997.

[8] J. Hertzberg and F. Kirchner. Landmark-based autonomous navigation in sewerage pipes. In *Proc. of the First Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

[9] H. Hu and M. Brady. A bayesian approach to real-time obstacle avoidance for a mobile robot. In *Autonomous Robots*, volume 1. Kluwer Academic Publishers, Boston, 1994.

[10] L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[11] M. Khatib and R. Chatila. An extended potential field approach for mobile robot sensor-based motions. In *Proc. International Conference on Intelligent Autonomous Systems (IAS'4)*, 1995.

[12] O. Khatib. Real-time obstacle avoidance for robot manipulator and mobile robots. *The International Journal of Robotics Research*, 5(1), 1986.

[13] S. Koenig and R.G. Simmons. A robot navigation architecture based on partially observable markov decision process models. In Kortenkamp et al. [14].

[14] D. Kortenkamp, R.P. Bonasso, and R. Murphy, editors. *Artificial Intelligence and Mobile Robots*. MIT/AAAI Press, Cambridge, MA, 1998.

[15] I. Nourbakhsh, R. Powers, and S. Birchfield. DERVISH an office-navigating robot. *AI Magazine*, 16(2), Summer 1995.

[16] R. Simmons. The curvature-velocity method for local obstacle avoidance. In *Proc. of the IEEE International Conference on Robotics and Automation*, 1996.

[17] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. of the International Joint Conference on Artificial Intelligence*, 1995.

[18] S. Thrun. Learning maps for indoor mobile robot navigation. *Journal of Artificial Intelligence*, to appear.

[19] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. In Kortenkamp et al. [14].