Real-world Robot Navigation amongst Deformable Obstacles

Barbara Frank Cyrill Stachniss Rüdiger Schmedding Matthias Teschner Wolfram Burgard

***Abstract*— In this paper, we consider the problem of mobile robots navigating in environments with non-rigid objects. Whereas robots can plan their paths more effectively when they utilize the information about the deformability of objects, they also need to consider the influence of the interaction with the deformable objects on their measurements during the execution of their navigation task. In this paper, we present a probabilistic approach to identify the measurements influenced by the deformable objects. Based on a learned statistics about the influence of the deformable objects on the measurements, the robot is able to perform a sensor-based collision avoidance of unforeseen objects. We present experiments carried out with a real robot that illustrate the practicability of our approach.**

I. INTRODUCTION

The ability to safely navigate in their environment is one of the fundamental tasks of mobile robots. Accordingly, the problem of safe navigation has received considerable attention in the past. The majority of approaches for navigation, however, has been developed for environments with rigid obstacles [17, 13] and does not consider the potential deformations imposed on the corresponding objects while the robot navigates through the environment. In the real world, however, not all obstacles are rigid and taking this knowledge into account can enable a robot to accomplish navigation tasks that otherwise cannot be carried out. For example, in our everyday life we often deal with deformable objects such as plants, curtains, or cloth and we are also able to utilize the information about the deformability of the corresponding objects. Consider, for example, the situation in which a curtain blocks a potential path of the robot as depicted in Fig. 1. Without the knowledge that the curtain can be deformed, the robot would always have to take a detour. Precise information about the cost of potential deformations, however, allows the robot to plan cost-optimal paths through the corridor, thereby deforming the curtain at minimal cost.

For robots that operate in environments with deformable objects, two tasks are essential. First, the robot needs to be able to take the cost of deformations resulting from its interaction with deformable objects into account during the path planning process. Furthermore, the robot needs to be able to appropriately interpret its sensory input during the interaction with the deformable objects. For example, during the interaction, the robot necessarily gets close to the deformable object so that its field of view might get obstructed. However, for safe navigation the robot still needs to be able to identify the measurements that do not correspond to the deformable object and come from other, possibly even rigid objects.

Fig. 1. The mobile robot Albert reasoning about its trajectory.

In this paper, we present a probabilistic approach that allows a mobile robot to distinguish measurements caused by deformable objects it is interacting with from ordinary measurements. This allows the robot to utilize standard reactive collision avoidance techniques like potential fields [12] or dynamic window techniques [4, 3, 14] simply by filtering out measurements that are caused by the objects the robot is interacting with. Additionally, the ability to reliably identify measurements not perceiving parts of the deformable object enables the robot to correctly interpret them also for the sake of collision avoidance. Our approach has been implemented on a real robot and evaluated in a collision avoidance task carried out while the robot interacts with a curtain. The results demonstrate that our approach allows the robot to safely avoid obstacles while it is interacting with a deformable object.

This paper is organized as follows. After discussing related work in the following section, we present in Section III an overview of our current navigation system for robots operating in environments with deformable objects. In Section IV, we describe how our robot estimates the cost of deforming objects and how it incorporates this information during the path planning process. Section V then contains our approach to determining which sensors measurements are influenced by the deformable object and how this information can be incorporated into the collision avoidance process. Finally, Section VI contains experimental results.

All authors are with the Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany.
{bfrank,stachnis,schmedd,burgard,teschner}@informatik.uni-freiburg.de
This work has partly been supported by the German Research Foundation (DFG) under contract number SFB/TR-8.

II. RELATED WORK

Most approaches to mobile robot path planning assume that the environment is static and that all objects are rigid [13, 11, 2]. In the last years, however, path planning techniques for deformable robots in static environments have been presented [7, 10].

In case objects in the environment are deformable, the underlying model for deformations and the model of the environment have a substantial influence on the accuracy of the estimated deformations as well as on the performance of the planner. There exist geometric approaches such as the free-form deformation (FFD) that can be computed efficiently, for example, the FFD method of Sederberg and Parry [19]. Physically motivated approaches use either mass-spring systems [16] or Finite element methods (FEMs) which reflect physical properties of the objects in a better way, see [8, 15].

Kavraki *et al.* [10] developed the f-PRM-Framework that is able to plan paths for flexible robots of simple geometric shapes such as surface patches [9] or simple volumetric elements [1]. They apply a mass-spring model and the planner selects the deformation of the robot that minimizes its deformation energy. Similar to this technique, Gayle *et al.* [7] presented an approach to path planning for a deformable robot that is based on PRMs. To achieve a more realistic simulation of deformations they add constraints for volume preservation to the mass-spring model of the robot.

In the context of collision avoidance, several successful methods have been presented. They are typically executed with a higher frequency compared to path planning, operate mainly on the sensor data itself with the task to ensure collision free motion of the robot. Such methods can roughly be divided into map-based approaches such as road-map or cell-decomposition techniques (see [13] for an extensive overview), and reactive, sensor-based approaches [4, 14, 20]. Such methods are designed to react to unforeseen obstacles but assume all objects to be rigid.

The techniques described by Fox *et al.* [5] as well as Schmidt and Azarm [18] combine the sensory information with a given map of the environment to deal with objects that cannot be detected with the robot's sensors. Brock and Kathib [3] presented an integration of path planning and reactive collision avoidance. There exist also methods that incorporate speed into the planning process in combination with collision avoidance [22].

The techniques mentioned above that are able to deal with deformable objects have been mainly used in simulations and *not on real robots*. When applying those techniques in the real world, a series of problems arise such as how to interpret the sensor data perceived by the robot while it is deforming an object as well as adaptation to the collision avoidance system.

Our planning system applies FEMs to compute object deformations. In order to perform the path planning task efficiently, we precompute potential deformations for a set of robot movements through the objects and estimate the costs by means of regression. This is based on our previous approach [6]. In contrast to [6], we realize in this paper a planning system on a real physical mobile robot and not only in simulation which requires a series of adaptations and new techniques for successfully planning paths in environments with deformable objects. This includes a sound way on how to interpret the sensor data a mobile robot perceives while deforming an object. Our approach allows for filtering the range data obtained with the robot's sensor to label beams that are reflected from a deformable objects. This, in turn, makes our technique orthogonal to other collision avoidance techniques and enables the robot to combine existing techniques with our method. Thus, we explicitly address these open issues and are able to deploy a real robot with the capability of safely moving though environments with deformable objects, leaving the world of simulation behind.

III. SYSTEM OVERVIEW

Our approach to mobile robot motion planning in real environments with deformable objects uses a typical layered architecture for realizing the navigation functionalities. Besides drivers for sensors and the robot, the hardware abstraction layer, etc., three key components are the

- path planning module, the
- collision avoidance module, and the
- localization module.

The path planning system computes trajectories that guide the robot to its desired goal location and is executed with rather low frequency. In contrast to that, a collision avoidance module operates with high frequency in order to avoid collisions with unforeseen and/or dynamic obstacles. Finally, the localization module runs Monte-Carlo localization keeping track of the robot's pose.

In the context of navigation in real environments with deformable objects the key questions are: (i) how to plan trajectories in the presence of such obstacles and (ii) how to interpret the sensor data so that the robot can distinguish between unforeseen obstacles to avoid and deformable objects, which is needed for collision avoidance as well as for localization.

A prerequisite to address these issues is an appropriate model of the environment. First, a traditional map (here grid map) is needed to represent static obstacles. Second, deformable objects need to be modeled. It is, however, significantly more complex to represent deformable objects since one needs to store the three-dimensional structure of the object as well as its elasticity parameters to allow for adequate simulation of deformations.

IV. ROBOT TRAJECTORY PLANNING CONSIDERING OBJECT DEFORMATIONS

A. Learning Deformation Cost Functions

To allow for efficient generation of trajectories for a mobile robot in environments with deformable objects, we build upon our recent work [6]. The key idea is to learn cost functions for the individual deformable objects parameterized by different trajectories leading to deformations. In order to carry out this task in an efficient manner, a physical simulation engine is used in a preprocessing step to calculate the corresponding cost functions. For making adequate predictions of the object deformations, we apply finite element methods to model the deformations.

Once a set of trajectories deforming an object is simulated in order to obtain the corresponding costs, these values can be used to approximate the deformation cost function. Our path planner then evaluates trajectories using $A^\star$ according to the cost function

$$C(path) = \alpha\, C_{def}(path) + (1-\alpha)\, C_{travel}(path), \quad (1)$$

where $\alpha \in [0,1]$ is a user-defined weighting coefficient that determines the trade-off between deformation and path costs.

Given our current implementation, the robot is able to answer path queries in typical indoor environments in less than 1 second – in contrast to several hours that would be needed if the deformation simulations were carried out at runtime. For further details, we refer the reader to our previous work [6]. It should be noted that our approach makes the assumption that there are no interactions between the different deformable objects and that they are fixed in the environment, such as curtains or (rather heavy) plants.

B. Object Reconstruction

Our previous work dealt with the path planning issues on an abstract level carried out only in simulation. We furthermore assumed that accurate 3D models incorporating the deformation parameters are known. In this work, we go a step further and also learn the 3D model of the objects. This is done by using a real mobile robot equipped with a laser range finder mounted on a pan-tilt unit.

The robot perceives 3D range scans of the object from different perspectives and generates a consistent 3D model by means of the iterative closest point (ICP) algorithm. For the simulation of deformable objects, a tetrahedral mesh is needed, which is reconstructed from the 3D model as shown in [21]. This method can handle un-orientable, non-manifold or damaged surfaces, and is therefore particularly suitable for the reconstruction from 3D scans. Based on the 3D scan, a signed distance field is computed where the set of voxels having negative sign represents the volume of the object. Next, a uniform axis-aligned grid is laid over the distance field. All cells outside the volume are discarded and the remaining cubical cells are split into tetrahedrons. Finally, a smoothing filter is applied to optimize the tetrahedral mesh (see Fig. 2 for example models). Deformations of objects are then computed using a linear relation between the forces and displacements q of the single elements (i.e. the tetrahedrons):

$$f = K(E,\nu)q \quad (2)$$

with stiffness matrix $K(E,\nu)$ depending on the elasticity parameters Poisson ratio ν and Young modulus E.

One open issue is the question of how to determine the elasticity parameters of the individual objects after acquiring the 3D model. In our current system, these parameters are

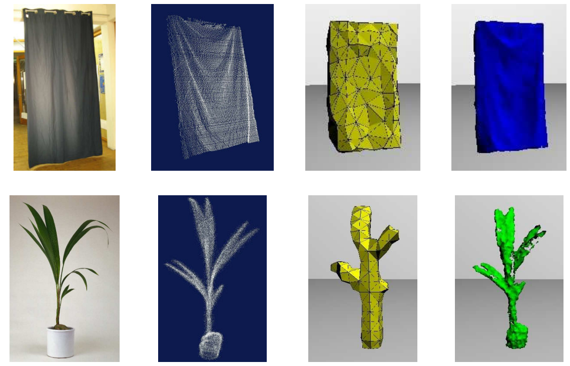

Fig. 2. Generating a model of a curtain (top) and a plant (bottom) for predicting the deformation cost: Left: photo. Second from left: point cloud. Second from right: tetrahedral mesh. Right: 3D model .

set manually. However, in a future step, we plan to acquire this information autonomously by the robot itself from force-displacement relations obtained with a 7-DoF manipulator. By applying a force to unknown objects and by measuring the displacement, we hope to learn the elasticity parameters. Such a procedure, however, is not yet implemented in our current system.

V. COLLISION AVOIDANCE

In this section, we describe the collision avoidance system developed for our robot that navigates in environments with deformable objects. Our robot is equipped with a SICK laser scanner with 180 degree opening angle. We use the range measurements for a basic collision avoidance behavior.

When navigating autonomously, the robot constantly has to observe its environment in order to react to unforeseen obstacles. At the same time, it might get close to deformable objects when deforming them. Therefore, the main problem in our setting is to figure out which measurements correspond to a deformable object, which means that these measurements can be ignored by the collision avoidance system. Note that we do not claim that our approach can distinguish deformable from rigid objects only based on laser data in general. However, by combining the knowledge about objects in the environment and their geometry with estimates of range scans during deformations, we can estimate the deformability of an observed object.

We model this problem in a probabilistic fashion: Let c_i denote the binary random variable which describes the event that beam i hits a deformable object. Then, $p(c_i \mid x, z_i)$ describes the probability that beam i hits a deformable object given the robot position x and the range measurement z_i. Applying Bayes' formula, we obtain

$$p(c_i|x,z_i) = \frac{p(z_i|x,c_i)p(c_i|x)}{p(z_i|x,c_i)p(c_i|x) + p(z_i|x,\neg c_i)p(\neg c_i|x)}. \quad (3)$$

Here, $p(z_i \mid x, c_i)$ is the sensor model and $p(c_i \mid x)$ is the prior denoting the probability of observing a deformable object from position x. We will shortly go into detail of how to learn these models. The sensor model $p(z_i \mid x, \neg c_i)$ corresponds to the common sensor model $p(z_i \mid x)$ when no deformable objects are present.

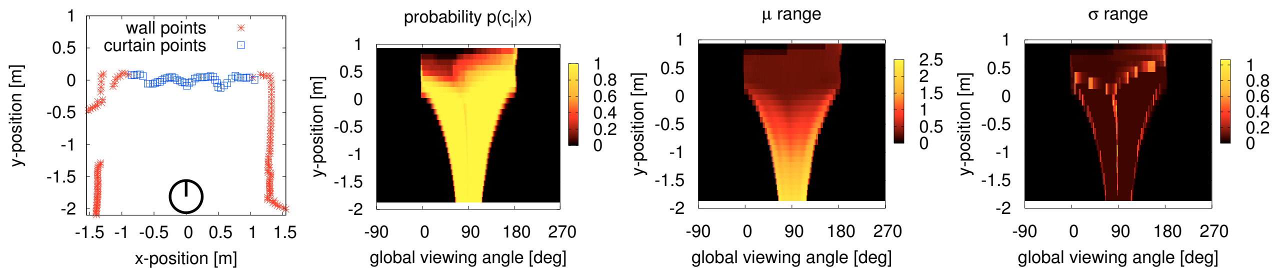

Fig. 3. Sensor model for the trajectory depicted in the left figure: shown are the probabilities $p(c_i \mid x)$ (second from left), the average beam length when observing the deformable object (second from right) as well as the standard deviation σ (right) for the robot position and the viewing angle.

A. Learning sensor models for deformable objects

The sensor model $p(z_i \mid x, c_i)$ not only depends on the robot position but also on the trajectory relative to an object. For instance, the robot will measure a different distance to the curtain when it is situated in front of it than it would while passing through and deforming the curtain. Therefore, we determine sensor models corresponding to different trajectories of the robot relative to an object.

For each trajectory, we record different datasets consisting of the robot positions x (provided by the localization module) and the ranges z_i and then manually label the beams reflected by the deformable object. From the labeled measurements obtained along these trajectories, we compute the statistics

$$p(c_i \mid x) = \frac{hits_{def}}{hits_{def} + misses_{def}}, \tag{4}$$

where $hits_{def}$ is the number of beams reflected by a deformable object and $misses_{def}$ states, how often no deformable object was observed for position x and viewing angle i. The sensor model $p(z_i \mid x, c_i)$ is described by a Gaussian with average range μ and variance σ^2. An example of the deformable sensor model for a typical robot trajectory through the curtain is shown in Fig. 3.

B. Avoiding collisions

During path execution, the robot constantly monitors its position and also its sensor measurements for utilization in the collision avoidance system. In our case, the robot has to distinguish between allowed collisions with deformable objects and impending collisions with unforeseen or dynamic obstacles which have to be avoided. This is done by filtering out the range measurements that observe a deformable object with high probability. Therefore, we evaluate Eq. (3) for each beam and identify those beams that can be neglected for the collision avoidance.

Note that this labeling or filtering of the range measurement offers a great potential since it is done orthogonal to traditional collision avoidance methods. As a result, this technique can be combined with any other collision avoidance technique as, for example, with the dynamic window approach [4] or the nearness diagram technique [14].

The detected measurements which are identified as belonging to dynamic obstacles can be incorporated into the navigation system to update the path of the robot or into any existing sensor based collision avoidance routine. Our current implementation performs replanning if a path is blocked by a dynamic object or simply stops the robot if the distance to an obstacle is too close. An example of the collision detection is given in Fig. 4.

VI. EXPERIMENTAL RESULTS

We performed several experiments to evaluate the performance of our developed planning system on a real robot. We used an iRobot B21r platform equipped with a SICK laser range finder. Our implementation is based on CARMEN which is a navigation software allowing independent modules to communicate via a middle-ware. To integrate our approach into CARMEN, we replaced the collision detection method inside the module "robot" as well as the planning module termed "navigator" with our software. In addition to that, we extended the localization module which is based on MCL so that the laser beams hitting a deformable object during deformation are not considered in the sensor model.

We mounted a set of curtains in the corridor of our lab as deformable objects. First, we evaluate our sensor model for deformable objects. Next, we analyze the performance of our collision avoidance system during path execution in the presence of unforeseen and dynamic obstacles. Finally, we give some examples of how the incorporation of the deformation cost function influences the path search.

A. Sensor model prediction

In the first experiment, we evaluated how well our sensor model for deformable objects is able to predict the presence of deformable objects. We learned a sensor model for two different trajectories through the curtain that were chosen preferably by our path planner. To compute the sensor model statistics, we recorded the laser data and the robot position and manually labeled the laser beams that were reflected by the curtain. For each trajectory, we performed a leave-one-out cross-validation using 11 trajectories for learning the model and one for evaluation. The results of this experiment are summarized in Table I and demonstrate, that the system is able to distinguish between deformable and static obstacles with high accuracy. While the number of false positives is at around 3%, the number of false negatives is below 1%.

B. Recognition of dynamic obstacles

While it is intuitive that the sensor model is able to distinguish between deformable and static obstacles, it is not clear how well the classification works in the presence of dynamic obstacles in the vicinity of the deformable

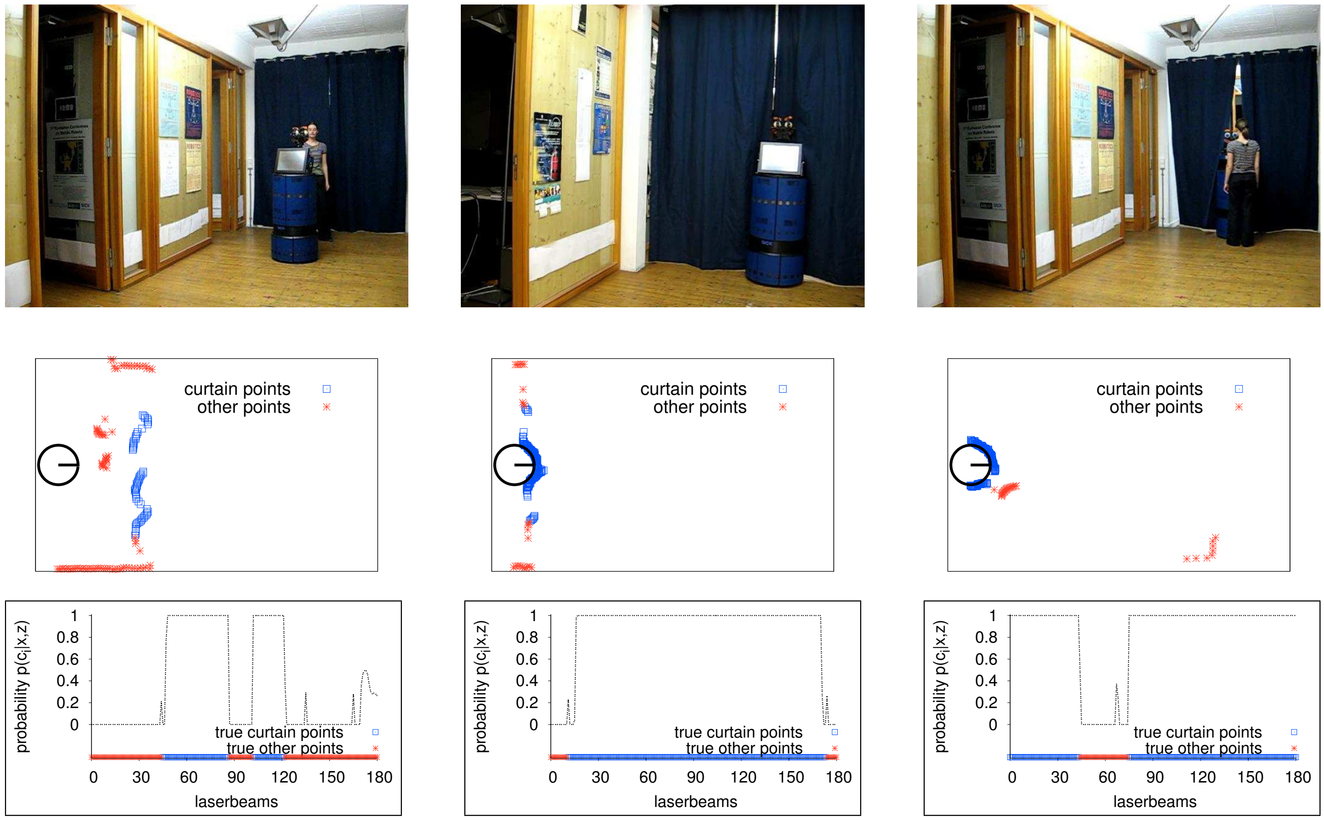

Fig. 4. Different collision avoidance scenarios (top row): Laser beams are evaluated with respect to their likelihood of observing a deformable object. In the second row, the classification of the individual laser beams is illustrated while in the bottom row, the probability of each beam together with the ground truth is shown.

	True label	
Detected label	Deformable Object	No deformable Object
Deformable Object	43857 (97.1%)	621 (0.9%)
No Deformable Object	1292 (2.9%)	65907 (99.1%)
Total	45149	66528

TABLE I

CONFUSION MATRIX FOR PREDICTING WHETHER A BEAM HITS A DEFORMABLE OBJECT IN A STATIC ENVIRONMENT.

	True label	
Detected label	Deformable Object	Dynamic Object
Deformable Object	8563 (96.5%)	98 (2.1%)
Dynamic Object	314 (3.5%)	4600 (97.9%)
Total	8877	4698

TABLE II

CONFUSION MATRIX FOR AN ENVIRONMENT CONTAINING BOTH DEFORMABLE AND DYNAMIC OBJECTS.

objects. The key question is whether the system is able to distinguish well between these obstacle classes and therefore is able to navigate safely. An important precondition for this is of course that the sensor can perceive a dynamic obstacle and that it is not completely occluded by the deformable object. To answer this question we performed several experiments where our robot moved on a trajectory deforming the curtain while dynamic obstacles were blocking its path. The recorded laser scans were labeled accordingly and evaluated with respect to the prediction performance. The results are listed in Table II. In this experiment, the number of false negatives is comparable to the situation in static environments while the number of false positives is around 1% higher than in the previous experiment. Our experiments, however, showed that this still leads to a safe behavior. In the worst case, the false negatives forced the robot to stop when it was not necessary while the false positives usually where outliers in a region of correctly classified dynamic obstacle beams. Therefore, the robot was still able to recognize dynamic obstacles and thus avoided collisions with these obstacles.

C. Example Trajectories through Curtains

For our experimental setup, we varied the trade-off between the deformation cost and the travel cost. The results for an example trajectory can be seen in Fig. 5. When the weighting coefficient α, which determines the trade-off between deformation and travel cost, is set to moderate values, then the planner prefers trajectories going through easily deformable objects. Note that in our scene, the curtain consists of two individual, neighboring curtains. The minimal-cost path, therefore, guides the robot through the contact point of both curtains. This fact can be observed in Fig. 6, where the curtains are moved compared to the previous example. Here, the planner chooses a slightly longer trajectory in order to minimize the deformation costs. Finally, a sequence of snapshots of our real robot navigating through the curtains is shown in Fig. 7. The execution of this

Fig. 7. The mobile robot Albert moving through a curtain.

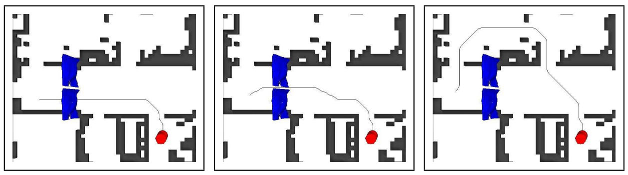

Fig. 5. Planning a trajectory for different weightings of the deformation cost ($\alpha = 0$ (left), $\alpha = 0.2$ (middle), $\alpha = 0.8$ (right)).

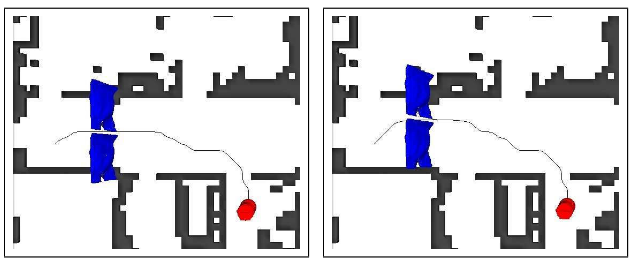

Fig. 6. The planner prefers trajectories that minimize object deformations. The curtains in the left picture are moved 40 cm along the positive y-axis compared to the picture on the right. The weighting coefficient α is set to 0.2 in both examples.

path together with demonstrations of the collision avoidance system can be found in the accompanying video.

VII. CONCLUSIONS

In this paper, we presented an approach for navigation in environments with deformable objects that explicitly takes into account the influence of the interaction between the robot and the deformable objects onto the measurements. Our approach is purely probabilistic and estimates for each measurement as to whether or not it might be caused by the deformable object in the environment. This allows the robot to get close to deformable objects and still avoid collisions with non-deformable objects. In our planning system, the costs of object deformations are determined using finite element methods to appropriately model the physical properties. Additionally we perform pre-computations to allow for an efficient online-calculation of path queries.

Our approach has been implemented and tested on a real robot and in a practical experiment, in which the robot is able to deform objects and at the same time avoid collisions with people. Future work will include the learning of the parameters of the deformable object based on the interaction between the robot and the objects so that better statistics about the influence on the sensory input can be calculated.

REFERENCES

[1] E. Anshelevich, S. Owens, F. Lamiraux, and L.E. Kavraki. Deformable volumes in path planning applications. In *Proc. of the Int. Conf. on Robotics & Automation (ICRA)*, pages 2290–2295, 2000.

[2] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man and Cybernetics*, 22(2):224–241, 1992.

[3] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *Proc. of the Int. Conf. on Robotics & Automation (ICRA)*, 1999.

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

[5] D. Fox, W. Burgard, and S. Thrun. A hybrid collision avoidance method for mobile robots. In *Proc. of the Int. Conf. on Robotics & Automation (ICRA)*, 1998.

[6] B. Frank, M. Becker, C. Stachniss, M. Teschner, and W. Burgard. Efficient path planning for mobile robots in environments with deformable objects. In *Proc. of the Int. Conf. on Robotics & Automation (ICRA)*, 2008.

[7] R. Gayle, P. Segars, M.C. Lin, and D. Manocha. Path planning for deformable robots in complex environments. In *Proc. of Robotics: Science and Systems (RSS)*, pages 225–232, 2005.

[8] M. Hauth and W. Strasser. Corotational Simulation of Deformable Solids. In *WSCG*, pages 137–145, 2004.

[9] C. Holleman, L.E. Kavraki, and J. Warren. Planning paths for a flexible surface patch. In *Proc. of the Int. Conf. on Robotics & Automation (ICRA)*, pages 21–26, 1998.

[10] L.E. Kavraki, F. Lamiraux, and C. Holleman. Towards planning for elastic objects. In *Robotics: The Algorithmic Perspective*, pages 313–325. A.K. Peters, 1998. Proc. of the Third Workshop on the Algorithmic Foundations of Robotics (WAFR).

[11] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

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

[13] J.-C. Latombe. *Robot Motion Planning*. Kluwer Academic Pub., 1991.

[14] J. Minguez and L. Montano. Nearness diagram navigation (nd): A new real time collision avoidance approach. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2094–2100, 2000.

[15] M. Mueller and M. Gross. Interactive Virtual Materials. In *Graphics Interface*, pages 239–246, 2004.

[16] A. Nealen, M. Mueller, R. Keiser, E. Boxerman, and M. Carlson. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum*, 25(4):809–836, 2006.

[17] N.J. Nilsson. A mobile automation: An application of artificial intelligence techniques. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 1969.

[18] K. Schmidt and K. Azarm. Mobile robot navigation in a dynamic world using an unsteady diffusion equation strategy. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, 1992.

[19] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. In *Proc. of the Conf. on Computer graphics and interactive techniques*, pages 151–160, 1986.

[20] R. Simmons. The curvature-velocity method for local obstacle avoidance. In *Proc. of the Int. Conf. on Robotics & Automation (ICRA)*, 1996.

[21] J. Spillmann, M. Wagner, and M. Teschner. Robust tetrahedral meshing of triangle soups. In *Proc. Vision, Modeling, Visualization (VMV)*, pages 9–16, 2006.

[22] C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 508–513, 2002.