2012 12th IEEE-RAS International Conference on Humanoid Robots
Nov.29-Dec.1, 2012. Business Innovation Center Osaka, Japan

Real-Time Navigation in 3D Environments Based on Depth Camera Data

Daniel Maier Armin Hornung Maren Bennewitz

***Abstract*—In this paper, we present an integrated approach for robot localization, obstacle mapping, and path planning in 3D environments based on data of an onboard consumer-level depth camera. We rely on state-of-the-art techniques for environment modeling and localization, which we extend for depth camera data. We thoroughly evaluated our system with a Nao humanoid equipped with an Asus Xtion Pro Live depth camera on top of the humanoid's head and present navigation experiments in a multi-level environment containing static and non-static obstacles. Our approach performs in real-time, maintains a 3D environment representation, and estimates the robot's pose in 6D. As our results demonstrate, the depth camera is well-suited for robust localization and reliable obstacle avoidance in complex indoor environments.**

I. Introduction

Autonomous robots are designed with the ulterior motives that at one point, they can assists humans with tasks such as home-care, delivery, etc. All of these high-level tasks require that the robot is able to localize itself in the environment, detect obstacles, and avoid collisions with them by keeping track of their locations and planning collision-free paths around them. For localization and obstacle detection, an autonomous robot has to rely on onboard sensor information. Numerous sensors have been used for this purpose, including ultrasound sensors, laser range finders, as well as monocular and stereo cameras. All of these sensors suffer from shortcomings such as inaccuracy, sparseness, high algorithmic complexity, or simply weight or cost. Recently, depth cameras operating with projected infrared patterns such as the Microsoft Kinect or Asus Xtion series have become available on the consumer market, lifting some of these limitations. These cameras are relatively accurate and provide dense, three-dimensional information directly from the hardware. To the best of our knowledge, in this paper, we present the first integrated navigation system consisting of localization, obstacle mapping, and collision avoidance for humanoid robots that is based on depth camera data.

For a humanoid robot acting in complex indoor environments containing multiple levels and 3D obstacles, a volumetric representation of the environment is needed. Our approach relies on a given 3D environment model in form of an octree [1] that contains the static parts of the environment. In this representation, the robot estimates its pose using Monte Carlo localization based on acquired depth camera data. Given the estimated 6D pose of the humanoid and a sequence of depth images, our system continuously builds a local 3D representation of the current state of the environment containing also non-static obstacles. This learned octree-based representation is then used for real-time planning of collision-free paths.

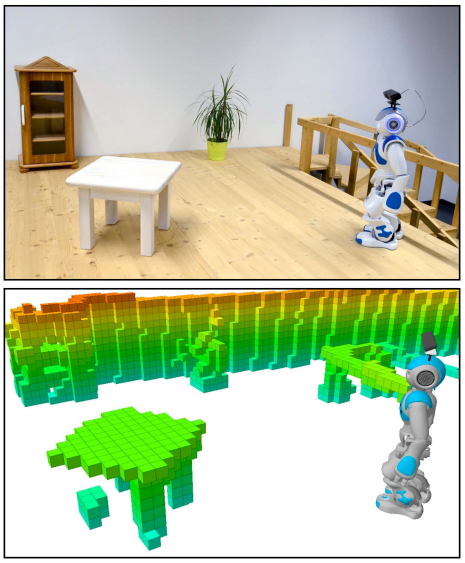

Fig. 1. Top: Nao humanoid robot with a depth camera on its head and part of the multi-level environment. Bottom: 3D representation of the scene used for collision avoidance. The map was constructed in real-time by turning on the spot for about 60°, thereby integrating 28 depth images.

Fig. 1 shows a motivating example of our system. The upper image depicts a humanoid navigating on the top level of a two-story environment. The lower image shows the robot's internal representation of its pose estimate and the local environment model, both maintained from depth camera measurements. In the environment model, one can clearly identify objects such as the table, the cabinet, the plant, or parts of the railing. The map was constructed in real-time by turning on the spot for about 60°, thereby integrating 28 depth images.

After presenting the basic techniques and our extension towards depth camera data, we illustrate the performance of our system for a Nao humanoid equipped with a Asus Xtion Pro Live sensor on top of the head. During the experiments, the robot navigated in a 3D environment consisting of multiple levels and containing several static and non-static obstacles. We thoroughly evaluate our approach and show results that demonstrate that our system leads to robust

All authors are with the Humanoid Robots Lab, University of Freiburg, Germany. This work has been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 and within the Research Training Group 1103. Their support is gratefully acknowledged.

localization and reliable obstacle avoidance in real-time. We conclude that such consumer-level depth cameras are well-suited for reliable humanoid robot navigation in complex indoor environments.

II. Related Work

The work most closely related to our approach has been recently presented by Biswas and Veloso. The authors developed an approach for indoor robot navigation based on depth camera data [2]. They proposed to sample points from the depth data belonging to vertical planes. These points are down-projected to 2D and used to update the particle filter that estimates the robot's pose. The observation model hereby matches the projected points to a given map of walls. The projected points are further used for obstacle detection. One disadvantage of this approach is that it discards the 3D information of the sensor data. Therefore, robots using this techniques cannot navigate in multi-level environments.

Hornung *et al.* presented a 3D localization method for humanoid robots based on 2D laser data [3]. Similar to our method, they applied a particle filter to estimate the 6D pose of the robot in a given 3D volumetric map of the environment. Our work can be seen as an extension as our approach does not require an expensive laser range finder but uses comparably cheap depth cameras. Furthermore, our system additionally contains 3D obstacle mapping and path planning capabilities.

Baudouin *et al.* proposed an approach to footstep planning and collision avoidance in 3D environments [4]. While the approach works in real-time and allows the robot to step over low obstacles, it relies on very accurate off-board sensing and applies a sampling technique for path planning that can result in arbitrarily suboptimal paths.

Nakhaei and Lamiraux presented a technique to 3D environment modeling from stereo data for humanoid motion planning [5]. Similar to our approach, the authors proposed to use a probabilistic voxel grid. However, their system has no localization component, which leads to inconsistencies in the learned map.

Ozawa *et al.* developed a system that relies on stereo image sequences to construct a dense local feature map [6]. This system performs real-time mapping with a humanoid robot based on 3D visual odometry for short trajectories.

Pretto *et al.* estimate the 6D pose of a humanoid as well as the 3D position of features in monocular camera data [7]. The authors designed feature detectors specifically to be able to deal with the effect of motion blur that typically occurs during humanoid walking. However, because the detected features in the monocular image are sparse, the approach is unsuitable for reliable obstacle detection.

Einhorn *et al.* presented an approach to 3D scene reconstruction and obstacle detection based on monocular vision [8]. The authors propose to track features in consecutive images and recover the features' positions from the ego-motion of the camera. This requires an accurate estimate of the camera pose, which the authors obtain from odometry of a wheeled robot. The system also relies on sparse features for obstacle detection.

Chestnutt *et al.* use 3D laser data acquired with a constantly sweeping scanner mounted on a pan-tilt unit at the humanoid's hip [9]. The authors fit planes through 3D point clouds and construct a 2.5D height map of the environment. Afterwards, they distinguish between accessible areas and obstacles based on the height difference. Such a sensor setup can only be used on robots with a significantly larger payload than the Nao humanoid. Gutmann *et al.* also build a 2.5D height map given accurate stereo data and additionally update a 3D occupancy grid map to plan navigation paths for the robot [10].

Kümmerle *et al.* developed a laser-based localization system for so-called multi-level surface maps for wheeled robots [11]. These maps store multiple levels of the scene per 2D grid cell and compactly represent 3D environments. However, they suffer from the disadvantage, that they do not provide volumetric information which is needed for humanoid navigation.

Stachniss *et al.* presented a simultaneous localization and mapping system (SLAM) to learn accurate 2D grid maps of large environments with a humanoid equipped with a laser scanner located in the neck [12]. Such a map was subsequently used by Faber *et al.* for humanoid localization and path planning in 2D [13]. During navigation, the robot avoids obstacles sensed with the laser scanner and ultrasound sensors located at the hip. Obstacles with a lower height are not detected which potentially leads to collisions. Also Tellez *et al.* use laser data to construct a 2D occupancy grid map in which paths for a humanoid are planned [14]. The authors use data from two laser scanners mounted on the robot's feet. All these approaches insufficiently represent the environment for navigation tasks in indoor scenarios with complex 3D structures.

Recently, approaches have been presented that perform SLAM with RGB-D cameras [15, 16, 17]. These approaches are optimized for small workspaces such as desktops or small rooms but not for larger environments. Further, they are algorithmically challenging and require that the camera can see enough texture or structure to match the observations. Consequently, they are not appropriate for scenarios like ours, where the camera faces the lowly-textured floor most of the time, in order to sense obstacles in the robot's way.

III. Navigation Based on Depth Camera Data

In this section we describe our approach to robot localization, mapping, and path planning.

A. Environment Representation

To enable modeling of multi-level environments containing obstacles of various shapes we use the octree-based mapping framework *OctoMap* [1]. This map representation partitions the space into free and occupied voxels where each voxel is associated with an occupancy probability. Unknown space is implicitly modeled by missing information in the

Fig. 2. Photograph of the environment in which we carried out the experiments and the corresponding map constructed with a CAD software. The map contains only the static parts of the environment.

tree. As opposed to a fixed size voxel grid map, this tree-based approach allows the map to grow dynamically and is compact in memory as it only allocates memory as needed. Bounded occupancy values enable to appropriately react to changes over time and enable a compression by pruning the tree, particularly in the large free areas.

We use two different 3D maps. First, we consider a static map of the environment for localization and as prior knowledge for path planning. Fig. 2 shows an example map. Secondly, we maintain an additional map containing local obstacles, which is continuously updated based on the depth data acquired by the robot while walking. This representation is then used for path planning around non-static obstacles.

For this process, we maintain a projected 2D map for efficient collision checks as in [18]. Each 3D map update of the local obstacle map also updates the 2D projection. To allow the robot to pass below underpasses and traverse the upper level of the environment, only obstacles within the vertical extent of the robot are hereby projected into the 2D obstacle map. Further, we filter out points corresponding to the floor, prior to map updates. Therefore, we consider a point's normal from its local neighborhood in the point cloud constructed from the depth image.

B. Probabilistic 3D Map Update

We integrate sensor readings into the local map by using occupancy grid mapping in 3D as in [1]. The probability $P(n \mid z_{1:t})$ that voxel n is occupied at time t is recursively computed given all sensor measurements $z_{1:t}$ according to

$$P(n \mid z_{1:t}) = \left[1 + \frac{1 - P(n \mid z_t)}{P(n \mid z_t)} \frac{1 - P(n \mid z_{1:t-1})}{P(n \mid z_{1:t-1})} \frac{P(n)}{1 - P(n)}\right]^{-1}, \quad (1)$$

where z_t is the measurement, $P(n)$ is the prior probability (typically this value is assumed to be $P(n) = 0.5$), and $P(n \mid z_{1:t-1})$ is the previous estimate. The term $P(n \mid z_t)$ denotes the likelihood of voxel n being occupied given the measurement z_t. Here, we employ a beam-based inverse sensor model that assumes that endpoints of a measurement correspond to obstacle surfaces and that the line of sight between sensor origin and endpoint does not contain any obstacles. Thus, we update the last voxel on the beam as occupied, and all the others up to the last one as free and use corresponding likelihoods for $P(n \mid z_t)$. For efficiency, we use the log-odds formulation of (1) to update the map.

C. Localization

For localization in the 3D model, we extend the Monte Carlo localization (MCL) framework by Hornung *et al.* [3], which was originally developed for data of 2D laser range finders, to depth camera data. Hereby, the humanoid's 6D pose is tracked in the 3D world model. The humanoid's torso serves as its base reference frame.

The pose $\mathbf{x} = (x, y, z, \varphi, \theta, \psi)$ consists of the 3D position (x, y, z) with roll, pitch, and yaw angles (φ, θ, ψ). For robust localization while walking, we combine 3D range data from the depth camera located on top of the head, attitude data provided by an inertial measurement unit (IMU) in the chest, and odometry data.

Odometry is computed from measured joint angles with forward kinematics and integrated in MCL with a Gaussian motion model. In the observation model, we consider the data of the humanoid's sensors. The depth camera provides a depth image, that we convert to a set of beams with ranges $\mathbf{r}_t$, the joint encoders provide a measurement $\tilde{z}_t$ of the humanoid's torso above the current ground plane, and the IMU estimates the roll and pitch angles $\tilde{\varphi}_t$ and $\tilde{\theta}_t$.

We assume that all these measurements are independent and combine them into one unified observation model to compute the likelihood of an observation $\mathbf{o}_t$:

$$\begin{aligned} p(\mathbf{o}_t \mid \mathbf{x}_t) &= p(\mathbf{r}_t, \tilde{z}_t, \tilde{\varphi}_t, \tilde{\theta}_t \mid \mathbf{x}_t) = \\ &\quad p(\mathbf{r}_t \mid \mathbf{x}_t) \cdot p(\tilde{z}_t \mid \mathbf{x}_t) \cdot p(\tilde{\varphi}_t \mid \mathbf{x}_t) \cdot p(\tilde{\theta}_t \mid \mathbf{x}_t). \end{aligned} \quad (2)$$

Here, $\mathbf{x}_t$ is the robot's estimated state.

For evaluating the range sensing likelihood $p(\mathbf{r}_t \mid \mathbf{x}_t)$, we sample a sparse subset of beams from $\mathbf{r}_t$ (see below). We assume that the sampled measurements $r_{t,k}$ are conditionally independent and determine the likelihoods of the individual beams $p(r_{t,k} \mid \mathbf{x}_t)$ by ray casting in the volumetric 3D environment representation. Hereby, we extract for each beam the expected distance to the closest obstacle contained in the map, given the robot pose, and compare it with the actually measured distance. To evaluate the measurement and to model the measurement uncertainty of the sensor, we use a Gaussian distribution. Similarly, we integrate the torso height $\tilde{z}_t$ as well as the roll $\tilde{\varphi}_t$ and pitch $\tilde{\theta}_t$ provided by the IMU with a Gaussian distribution based on the measured values and the predicted ones.

To sample the beams $r_{t,k}$ in the ray casting step, our system classifies all end points of the beams $\mathbf{r}_t$ into ground and non-ground parts. Therefore, it pre-filters candidates based on their height in the robot's internal coordinate system. Then it obtains the beams hitting the ground by finding dominant planes with RANSAC over local neighborhood normals. Our system uses this information for uniformly sampling half the beams from non-ground parts and the other half from the ground. Thus, we compensate for the fact that the camera faces mostly the floor area for better obstacle avoidance. Beams hitting the floor, however, can provide no information for estimating translation in the horizontal plane, which is typically more important than height or pitch and roll.

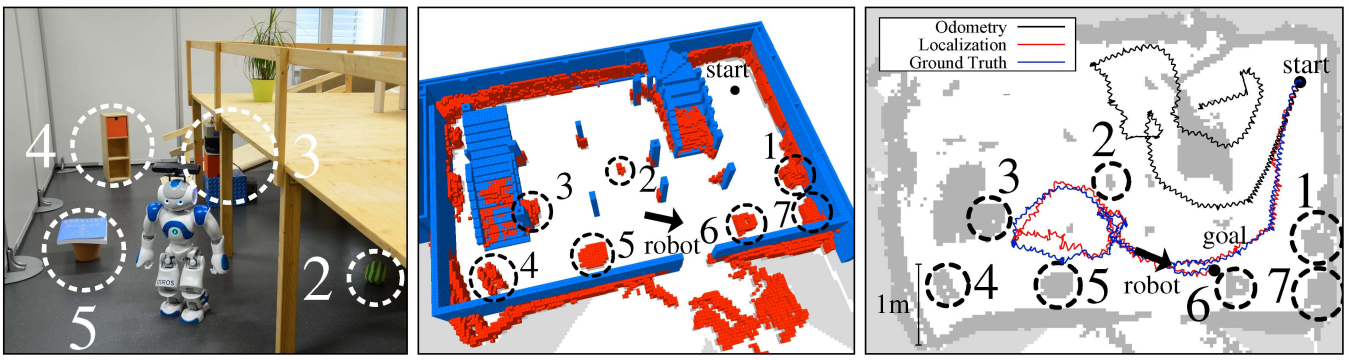

Fig. 3. Left: Nao navigating in the lower level of our environment between obstacles (Scenario 2). Middle: Static (blue) and local (red) 3D map constructed by the robot while walking. Right: Two-dimensional projection of the local map used for collision avoidance and path planning. The lines show the robot's odometry estimate, the estimated pose of our localization system, and the ground truth. The arrow indicates the robot's pose corresponding to the left image. The numbered circles in the figure indicate obstacles that are not part of the static map. As can be seen, our navigation system localizes the robot accurately and leads to reliable navigation behavior.

D. Path Planning and Collision Avoidance

For planning collision-free path, we consider the static map of the environment as prior knowledge, as well as the locally constructed map based on depth camera data, which contains also non-static obstacles.

In general, one could also plan collision-free footsteps or whole-body motions for the humanoid on the learned 3D map. However, planning motions in 3D is still a very complex problem and requires either high computation times or provides arbitrarily suboptimal solutions. For sake of real-time performance and robustness, we therefore rely on a projection of the 3D map to the floor. To be able to traverse underpasses, we restrict the projected area to the size of the robot in vertical direction. This is the area where collisions are potentially hazardous for the robot. Everything below and above can be safely ignored. Note that this is not the same as maintaining a simple 2D map. When the robot's z-coordinate changes, the projection is updated accordingly. This is only possible because we keep the 3D structure, hence enabling navigation in multi-level scenarios.

For collision checks in the projected map, we assume a circular robot model. This assumption prevents the robot from passing very narrow passages but allows to perform collision checks in constant time, once a distance transform of the 2D obstacle map is computed. These distance transforms can be computed in real-time.

To compute a collision-free path to the goal location, our system uses the A* algorithm. In case of a map update, it checks whether the previous plan is still valid and replans the path only if necessary.

IV. Experiments

We carried out a series of experiments demonstrating the capabilities of our navigation system based on depth camera data. All experiments were carried out with a Nao (V4) humanoid by Aldebaran Robotics. Nao is 58 cm in height, weighs 5.2 kg and has 25 degrees of freedom. With the current firmware of the robot, it is able to walk up to 10 cm/s. We modified the head and mounted an Asus Xtion Pro Live RGB-D camera on top of it (see Fig. 1). The camera has a field of view of 58° horizontally and 45° vertically. The camera is mounted on the robot's head in a way such that its optical axis faces the floor in a 30° angle while walking. We found this to be the best compromise between observing the near range for obstacle detection and looking ahead for localization and path planning. The increased weight due to the mounted camera destabilizes the walking behavior of the robot. We therefore added thin plastic sheets to the robot's feet to increase the friction.

To allow for real-time performance, we set the camera's resolution to 320×240 and update the map from sensor data at approximately 6 Hz. All processing is done on a standard quad core PC. We conducted the experiments in a multi-level environment, scaled-down to match the size of a Nao humanoid (see Fig. 2). We sketched the structure in a 3D CAD software and converted it to an *OctoMap*. This model is used for localization. Note that the 3D model does not perfectly match the actual scene due to imperfection in constructing the environment and, furthermore, the scene will contain non-static obstacles not included in the 3D CAD model. Therefore, our approach constructs a local map from depth camera data during navigation in real time. A video demonstrating our approach can be found at `http://hrl.informatik.uni-freiburg.de`.

A. Localization Accuracy

First, we performed a series of experiments to evaluate our localization system. We compared the resulting pose estimate to the ground truth in the 2D plane, which we obtained by tracking the humanoid with two stationary SICK laser range finders [19]. Consequently, we evaluated the translational error in the horizontal plane.

We conducted experiments in three different scenarios. In Scenario 1, the robot navigated on the lower level of our environment. Except for the two laser range finders used to record the ground truth and their power supplies, the static map closely resembled the actual scenario as can be seen in Fig. 2. Scenario 2 was similar to Scenario 1 but we

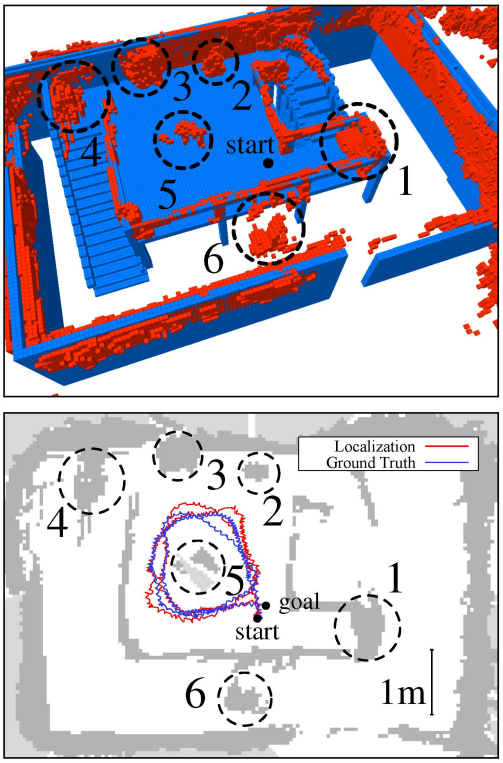

Fig. 4. Top: Local three-dimensional map (red) constructed while the robot was walking on the top level of the environment (Scenario 3), along with the static map (blue). Bottom: Two-dimensional projection of this map used for collision avoidance and path planning. As can be seen, also obstacles not contained in the original static representation are represented accordingly (numbered circles). The figure further shows the robot's trajectory estimated by our algorithm (red) and the ground truth (blue). Odometry has been left out for sakes of clarity.

additionally placed several obstacles such as books, balls, baskets, and shelves in the scene (left image in Fig. 3). These obstacles are not part of the map used for localization and therefore are expected to decrease the performance. The robot followed a similar path in Scenario 1 and 2 (right image in Fig. 3). In Scenario 3, the robot walked two circles on the top level of the environment where we also placed unmapped obstacles such as a table, a plant, and a cabinet in the scene (see Fig. 1).

For all scenarios, we manually initialized the localization system from the true pose. We used 500 particles for tracking the robot's pose and sampled 100 points from the depth image for ray casting as described in Sec. III-C. Fig. 3 and Fig. 4 depict the trajectories of the pose estimates and the ground truth on top of the projected obstacle map. As one can see, the estimated pose closely resembles the ground truth. Fig. 3, also shows the robot's odometry as reference which is clearly useless for reliably executing navigation tasks.

To evaluate the localization results quantitatively, we computed the mean error as well as the standard deviation over the robot's trajectory for each of the three scenarios. Table I summarizes the results. As can be seen, our system leads to robust and accurate pose tracking. As expected, the accuracy decreases slightly in Scenario 2 compared to Scenario 1, due

TABLE I
AGGREGATED LOCALIZATION ERROR FOR THREE SCENARIOS

	mean error [m]	std. dev. [m]
Scenario 1	0.07	0.04
Scenario 2	0.09	0.07
Scenario 3	0.07	0.05

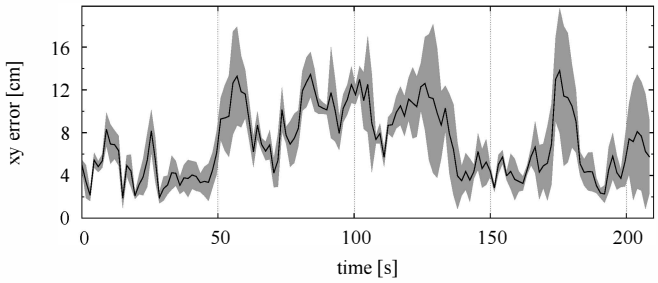

Fig. 5. Localization error as mean and standard deviation for five runs in scenario 1.

to the additional obstacles. However, for all scenarios, the average accuracy is still better than 1 decimeter.

Additionally, as Monte Carlo localization is a probabilistic technique, we repeated the accuracy evaluation five times over the same data set. Here, we used the same raw sensor data and initialization as for Scenario 1 and recorded the pose estimation errors relative to the ground truth. We then computed the mean error and standard deviation over the trajectory parametrized by time. Fig. 5 shows the results. The error is generally small. For the time between 50 s and 130 s and 170 s to 180 s, the average error increased from approximately 6 cm to 10 cm. Here, the robot navigated in the long hallway part parallel to the walls and the camera observed only little structure in the environment that could help reducing the translational uncertainty. A larger field of view or an active sensing approach could help in this case.

B. Mapping

To demonstrate the mapping and obstacle detection capabilities of our system, we consider Scenarios 2 and 3 described in the previous section. Fig. 3 and Fig. 4 show the 3D map constructed from the depth camera data while walking (red), as well as the static map of the environment (blue). It is clearly visible that the constructed map closely follows the structure of the reference map. Furthermore, it also includes all the obstacles that are not part of the static map. The figures also show the 2D projection of the constructed 3D map used for collision avoidance. In both maps, the structure of the non-traversable area is clearly identifiable.

C. Path Planning and Obstacle Avoidance

In the last experiment, we evaluate the ability to react to changes in the environment and plan collision-free paths also with non-static obstacles. Fig. 6 shows a scenario in which the robot reacted to a dynamic obstacle during walking. The left column shows the state of the projected obstacle map along with the robot's pose and the planned path. The

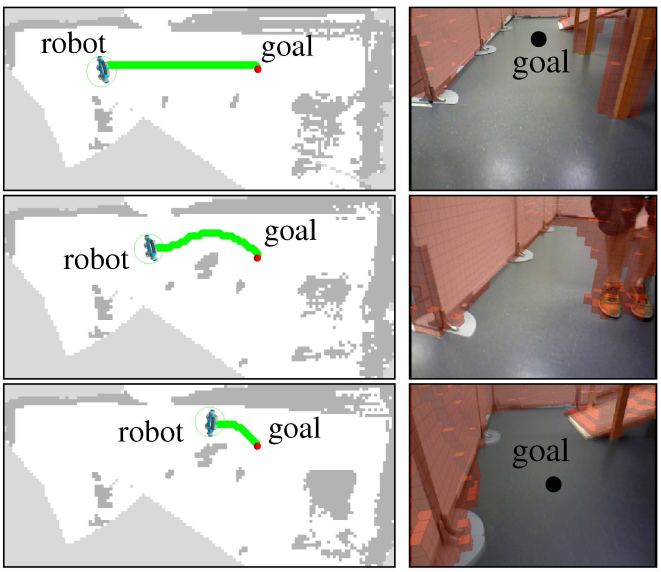

Fig. 6. The robot avoids a dynamic obstacle. The first row shows the robot's initial path to the goal with the corresponding camera image. Then, a human blocks the robot's path, forcing the robot to detour (second row). The robot follows the updated path to the goal (last row). The camera images show an overlay of the current obstacle map (red, best viewed in color).

right column depicts the current RGB camera image with an overlay of the state of the constructed 3D map. Initially, the robot planned a straight path to the goal location through the empty space (first row). While walking, a human entered the scene blocking the robot's initial path (second row). The robot immediately updated its obstacle map and planned a collision-free path to the goal. The robot followed that path accurately (third row) until it reached the goal.

V. Conclusions

In this paper, we demonstrated that affordable, consumer-level depth cameras are well-suited sensors for robot navigation tasks in complex indoor environments. We presented a real-time navigation system that allows to estimate a humanoid's 6D pose while walking and to map the scene in a local 3D map. We described how our system can be used for planning collision-free paths through scenes with static and non-static obstacles.

In experiments with a Nao humanoid equipped with an Asus Xtion Pro Live RGB-D camera, we thoroughly evaluated the performance of our system. As the results show, our approach leads to accurate localization estimates and reliable, collision-free navigation in the acquired 3D map. In the future, we will extend the approach to multi-level collision maps for different parts of the robot as in [18]. Hence, we will lift the circular robot model assumption made in this paper and allow the robot to better pass narrow passages.

Of course, depth cameras also have drawbacks. In the near range of the camera (closer than 50 cm), no depth data is available. In this case, we can fall back to applying collision detection approaches based on monocular vision data [20]. However, we rarely observed this problem in practice.

Acknowledgments

The authors would like to thank Mathias Luber for his help in laser-based tracking of the robot's pose for ground truth data during the experiments.

References

[1] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010. Software available at http://octomap.sf.net/.

[2] J. Biswas and M. Veloso. Depth camera based indoor mobile robot localization and navigation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[3] A. Hornung, K. M. Wurm, and M. Bennewitz. Humanoid robot localization in complex indoor environments. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[4] L. Baudouin, N. Perrin, T. Moulard, F. Lamiraux, O. Stasse, and E. Yoshida. Real-time replanning using 3d environment for humanoid robot. In *IEEE Int. Conf. on Humanoid Robots (Humanoids)*, 2011.

[5] A. Nakhaei and F. Lamiraux. Motion planning for humanoid robots in environments modeled by vision. In *IEEE Int. Conf. on Humanoid Robots (Humanoids)*, 2008.

[6] R. Ozawa, Y. Takaoka, Y. Kida, K. Nishiwaki, J. Chestnutt, J. Kuffner, S. Kagami, H. Mizoguchi, and H. Inoue. Using visual odometry to create 3d maps for online footstep planning. In *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, 2005.

[7] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello. A visual odometry framework robust to motion blur. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009.

[8] E. Einhorn, C. Schröter, and H.-M. Gross. Monocular scene reconstruction for reliable obstacle detection and robot navigation. In *European Conf. on Mobile Robots (ECMR)*, 2009.

[9] J. Chestnutt, Y. Takaoka, K. Suga, K. Nishiwaki, J. Kuffner, and S. Kagami. Biped navigation in rough environments using on-board sensing. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[10] J.-S. Gutmann, M. Fukuchi, and M. Fujita. 3D perception and environment map generation for humanoid robot navigation. *Int. Journal of Robotics Research (IJRR)*, 27(10):1117–1134, 2008.

[11] R. Kümmerle, R. Triebel, P. Pfaff, and W. Burgard. Monte Carlo localization in outdoor terrains using multilevel surface maps. *Journal of Field Robotics (JFR)*, 25:346–359, 2008.

[12] C. Stachniss, M. Bennewitz, G. Grisetti, S. Behnke, and W. Burgard. How to learn accurate grid maps with a humanoid. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008.

[13] F. Faber, M. Bennewitz, C. Eppner, A. Goeroeg, A. Gonsior, D. Joho, M. Schreiber, and S. Behnke. The humanoid museum tour guide Robotinho. In *18th IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2009.

[14] R. Tellez, F. Ferro, D. Mora, D. Pinyol, and D. Faconti. Autonomous humanoid navigation using laser and odometry data. In *IEEE Int. Conf. on Humanoid Robots (Humanoids)*, 2008.

[15] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D slam system. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[16] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Int. Symp. of Robotics Research (ISRR)*, 2011.

[17] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011.

[18] A. Hornung, M. Phillips, E. G. Jones, M. Bennewitz, M. Likhachev, and S. Chitta. Navigation in three-dimensional cluttered environments for mobile manipulation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[19] M. Luber, G. D. Tipaldi, and K. O. Arras. Place-dependent people tracking. *Int. Journal of Robotics Research (IJRR)*, 30(3):280–293, March 2011.

[20] D. Maier and M. Bennewitz. Appearance-based traversability classification in monocular images using iterative ground plane estimation. In *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.