

Real-Time Obstacle Detection and Avoidance in the Presence of Specular Surfaces Using an Active 3D Sensor

Brian Peasley and Stan Birchfield
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634
{bpeasley, stb}@clemson.edu

Abstract—This paper proposes a novel approach to obstacle detection and avoidance using a 3D sensor. We depart from the approach of previous researchers who use depth images from 3D sensors projected onto UV-disparity to detect obstacles. Instead, our approach relies on projecting 3D points onto the ground plane, which is estimated during a calibration step. A 2D occupancy map is then used to determine the presence of obstacles, from which translation and rotation velocities are computed to avoid the obstacles. Two innovations are introduced to overcome the limitations of the sensor: An *infinite pole approach* is proposed to hypothesize infinitely tall, thin obstacles when the sensor yields invalid readings, and a control strategy is adopted to turn the robot away from scenes that yield a high percentage of invalid readings. Together, these extensions enable the system to overcome the inherent limitations of the sensor. Experiments in a variety of environments, including dynamic objects, obstacles of varying heights, and dimly-lit conditions, show the ability of the system to perform robust obstacle avoidance in real time under realistic indoor conditions.

I. INTRODUCTION

Detecting and avoiding obstacles is an important problem in mobile robotics. In the parlance of Brooks' well-known subsumption architecture [4], obstacle avoidance is the lowest, or zeroth, level of competence, meaning it is the core functionality of a mobile robot system upon which everything else depends. If a robot can be made to avoid coming into contact with objects in the environment, then other higher-level capabilities can safely be incorporated into the system. Yet, despite decades of research and development on the topic, robust and reliable obstacle avoidance remains a delicate problem that is difficult to ensure.

The most common sensors for obstacle detection and avoidance have been laser range finders, sonars, and cameras. Each of these has its own strengths and weaknesses, as seen in Table I. Laser range finders, while providing a dense, accurate depth array, consume large amounts of power and are expensive. A ring of ultrasonic sonar sensors is more economical, but the resulting depth readings are more coarsely spaced and less accurate. Both approaches suffer from only providing readings within a horizontal plane parallel to the floor. Unlike sonars and laser scanners, cameras do not directly provide geometrical measurements of an environment, which must instead be inferred from the pixel data (a difficult problem). Due to the close spacing of pixels, a camera captures raw data at a high spatial resolution, but a multi-camera (e.g., stereo) system usually is only able

to provide depth estimates for a sparse set of matched pixels. An active 3D sensor, such as the Xbox Kinect, overcomes all of these limitations by providing a fulcrum-based (rather than planar) set of dense, accurate depth readings in real time. Moreover, the cost of such systems has recently dropped tremendously, making them practical for robotic systems.

	Speed	Cost	Power	Resolution	Planar
Laser	Slow	High	High	High	Yes
Sonar	Fast	Low	Low	Low	Yes
Single camera	Fast	Low	Low	High	No
Multiple cameras	Fast	Low	Low	Low	No
3D sensor	Fast	Low	Low	High	No

TABLE I

COMPARISON OF DIFFERENT SENSORS FOR OBSTACLE DETECTION.

In this paper we propose to overcome the deficiencies of current approaches for obstacle detection by using a 3D depth sensor. Such depth sensors have recently become widely available, enabling affordable, accurate 3D sensing. The sensor is mounted on the front of a mobile robot base, and the dense 3D point cloud is transformed to a bird's-eye occupancy map in order to detect the obstacles in the robot's immediate field of view. Two novel solutions are proposed to overcome weaknesses of the 3D sensor. To overcome invalid readings due to specular surfaces, a projection scheme is applied in which infinitely-tall phantom obstacles are placed at strategic locations. To overcome invalid readings due to obstacles being too close to the sensor, a control strategy is adopted to cause the robot to turn when the percentage of valid readings is insufficient. The combination of these two solutions enables robust, real-time obstacle detection and avoidance in an indoor environment. Moreover, the 3D sensor facilitates capabilities impossible with planar-based sensors, such as driving under obstacles if there is sufficient height, or refusing to drive under obstacles low to the ground. Experimental results are shown for a variety of challenging scenarios such as thin obstacles, reflective obstacles, close obstacles, and hanging obstacles, as well as dynamic and dark environments.

II. PREVIOUS WORK

A number of researchers over the years have used sonars [22] and lasers [3] for obstacle avoidance. Departing from the traditional sonar ring, Nourbakhsh et al. [19], [20] showed

that a non-conventional arrangement of sonars including angled sensors leads to more robust behavior and, in particular, prevents decapitation (i.e., collision with an object at a height above the ring of sensors). An active vision system with a vertical laser slit is presented in [21]. Proximity and vision sensors were combined to demonstrate the utility of potential field concepts in [12].

Because cameras do not directly yield depth measurements, a myriad of approaches to exploit visual information have been pursued by various researchers. Probably the most common and straightforward approach is to match pixels in stereo images, then to triangulate to recover depth [14], [1]. A related approach is to measure the flow field divergence from the optical flow in an image [18], or optical flow in an omnidirectional image [13]. A real-time system combining intensity edge and color outputs was developed by Lorigo et al. [15], and machine learning techniques were used to estimate real-time depth from a single image by Michels et al. [16]. Other researchers have explored the possibility of mapping 3D sensor depth data to UV-disparity to allow for non-planar surfaces [11], [9]. Our approach, in contrast, uses 3D point clouds projected to the ground plane.

A few researchers have used the Kinect sensor for obstacle detection and/or avoidance. A collision-avoidance system for human interaction with a robot manipulator mounted to a table is presented in [6] and a method to localize and navigate a building from a depth image is presented in [5]. A Kinect sensor has also been used for obstacle detection (but not avoidance) for a flying quadrotor craft [2]; in this system most of the data is removed rather than using the entire point cloud in order to speed computation. In more closely related work, a mobile robot equipped with a Kinect sensor detects and avoids obstacles by projecting the point cloud onto the ground plane [17]. Our approach differs in that we present a reactionary system that addresses specific limitations of the Kinect sensor.

III. THE KINECT SENSOR AND ITS LIMITATIONS

The Xbox Kinect is a 3D depth camera consisting of two cameras and a laser-based infrared (IR) projector. One of the cameras is a standard RGB camera, while the other camera is an IR camera which looks for a specific pattern projected onto a scene by the laser-based IR projector. An image of the sensor along with example output can be seen in Figure 1. This sensor calculates the disparity of each pixel by comparing the appearance of the projected pattern with the expected pattern at various depths.

There are two primary issues with using the Kinect sensor for obstacle detection. First, any object with reflective material such as shiny metal may prevent the reflected light from the IR projector from reaching the IR camera, thus causing invalid depth readings at those locations. Secondly, since the sensor requires triangulation between the IR projector and IR camera, which are separated in space, there is a blind spot up to approximately 0.4 meters directly in front of the sensor. Therefore, anything closer than this range will not be seen by the sensor, leading to invalid readings. When the robot is

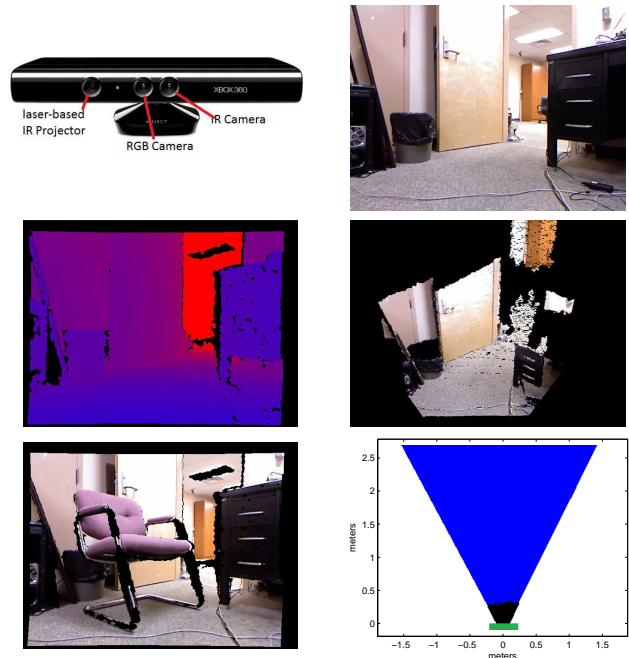


Fig. 1. The 3D Kinect sensor showing the two cameras and laser-based IR projector (top left). The sensor provides, in real time, an RGB color image (top right), a depth map (middle left), and a 3D point cloud obtained by combining the image and depth map (middle right). The two limitations of the sensor are the invalid readings obtained from specular surfaces (bottom left, shown in black) and the inability to obtain readings for obstacles closer than about 0.4 m right (bottom right, black indicates locations yielding invalid readings).

traveling toward a static object, this limitation is not of much concern, since the robot should turn away from the object before the limit is reached. However, in the case of dynamic obstacles that suddenly place themselves in front of the robot, or when the robot suddenly encounters a close obstacle due to a turn, such a blind spot can lead to a collision. Thankfully the invalid readings in both cases are not simply erroneous but rather are flagged as invalid, enabling post-processing algorithms to make educated guesses about the missing data. In later sections we propose solutions to both of these problems.

IV. APPROACH

The obstacle detection algorithm involves four main steps. First, the depth image from the Kinect sensor is transformed to a 3D point cloud using the calibration information embedded in the sensor. Secondly, the ground plane is segmented from the point cloud, and points belonging to the ground plane are removed. Thirdly, a 2D occupancy map is created by projecting the points to a top-down view. Finally, an obstacle avoidance control algorithm uses the occupancy map to decide how to move the robot. We now describe these steps in greater detail.

A. Floor Plane Estimation and Segmentation

Our approach is designed for indoor environments and therefore makes use of the so-called *ground plane constraint*, namely, that the robot moves on a flat ground plane. All

points that are above the ground plane are potential obstacles. The position and pitch of the sensor with respect to the robot base is assumed fixed, so that calculating the ground plane parameters involves a simple calibration step. The equation of the ground plane, in the camera coordinate system, can be modeled as

$$p_z = -\alpha p_x - \beta p_y - \delta, \quad (1)$$

where (p_x, p_y, p_z) are the 3D coordinates of a point on the ground plane. We assume that the area of the floor just in front of the sensor remains clear of obstacles during the calibration, which requires simply capturing a single instantaneous point cloud. Given n 3D points $\mathbf{p}^{(i)} = (p_x^{(i)}, p_y^{(i)}, p_z^{(i)})$, $i = 1, \dots, n$, in this uncluttered region, the parameters of the plane can be found by solving the following least squares problem:

$$\underbrace{\begin{bmatrix} -p_x^{(1)} & -p_y^{(1)} & -1 \\ -p_x^{(2)} & -p_y^{(2)} & -1 \\ \vdots & \vdots & \vdots \\ -p_x^{(n)} & -p_y^{(n)} & -1 \end{bmatrix}}_A \begin{bmatrix} \alpha \\ \beta \\ \delta \end{bmatrix} = \underbrace{\begin{bmatrix} p_z^{(1)} \\ p_z^{(2)} \\ \vdots \\ p_z^{(n)} \end{bmatrix}}_b \quad (2)$$

or

$$\begin{bmatrix} \alpha \\ \beta \\ \delta \end{bmatrix} = (A^T A)^{-1} A^T \mathbf{b}. \quad (3)$$

Once α , β , and δ have been determined, the signed distance from any point $\mathbf{p}^{(i)}$ to the plane, or equivalently, the height of the point above the ground plane, is given by

$$h^{(i)} = \frac{\alpha p_x^{(i)} + \beta p_y^{(i)} + p_z^{(i)} + \delta}{\sqrt{\alpha^2 + \beta^2}}. \quad (4)$$

If $h^{(i)} > 0$, then the i th point is above the ground plane, otherwise it is on or below the plane. To account for noise and harmless flat objects resting on the floor (e.g., a piece of paper), we declare a point to be above the ground plane only if it is greater than a threshold: $h^{(i)} > \tau$, where we set $\tau = 5$ cm.

B. Map Construction

One of the significant advantages of a 3D sensor over a laser or sonar is the ability to examine the height of obstacles. After segmenting the ground plane, the remaining points are potential obstacles, but some of these potential obstacles will be too high above the ground to cause the robot concern. If $h^{(i)} > \tau_h$, where $\tau_h = 0.5$ meters in our case, then the robot is able to pass under the point without collision.

The remaining points, i.e., those that are not on the ground plane but are low enough to cause collision, are transformed into a top-down coordinate system to yield a localized 2D map of the environment. We call these points the *obstacle points*. The 2D map is an orthographic projection of these obstacle points along the normal to the ground plane. To

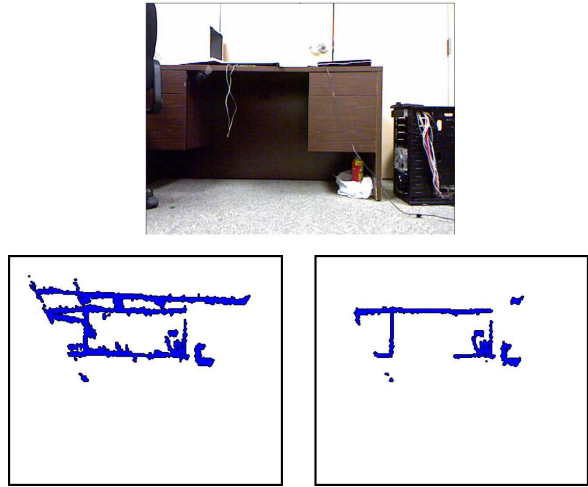


Fig. 2. TOP: An RGB image from the sensor. BOTTOM-LEFT: The occupancy map obtained by projecting all the non-floor points to a top-down coordinate system. Note that the desk is impassable in the map, even though the robot is short enough to pass under it. BOTTOM-RIGHT: The occupancy map obtained by projecting only the obstacle points below τ_h .

transform an obstacle point, $\mathbf{p}^{(i)}$, into the 2D ground coordinate system, we calculate

$$\mathbf{o}^{(i)} = \mathbf{p}^{(i)} - h^{(i)} \mathbf{n}, \quad (5)$$

where \mathbf{n} is the normal to the ground plane. Similarly, the point $\mathbf{c} = -h^{(i)} \mathbf{n}$ yields the projection of the sensors center of projection, $[0 \ 0 \ 0]^T$.

A binary occupancy map is created, where each obstacle point $\mathbf{p}^{(i)}$ causes the cell at (x, y) to be set to 1, where $x = d^{(i)} \cos \theta / \rho_x$ and $y = d^{(i)} \sin \theta / \rho_y$, where

$$d^{(i)} = \|\mathbf{c} - \mathbf{o}^{(i)}\| \quad (6)$$

$$\theta = \arctan \left(\frac{o_y^{(i)} - c_y}{o_x^{(i)} - c_x} \right), \quad (7)$$

where $\rho_x = \rho_y = 12.5$ cm is the spatial resolution of the grid. An example occupancy map is shown in Figure 2, where the importance of taking the height information into account is seen; otherwise the robot would not be allowed to pass under the desk even though there is enough head space for it to do so.

C. Handling Reflective Materials and the Blind Spot

We now describe two innovations for overcoming the limitations of the Kinect sensor. Because the sensor uses a laser-based IR projector to obtain a depth map of the environment, certain types of materials (such as glass or shiny metal) can cause invalid depth readings. For an obstacle avoidance system, it is crucial not to wander into regions in which there is no information. On the other hand, the overly conservative approach of never driving in a direction containing an invalid region would prevent the robot from, for example, driving toward a window even though there may be considerable free space before it reaches the window. Our approach achieves a safe compromise between the two extremes.

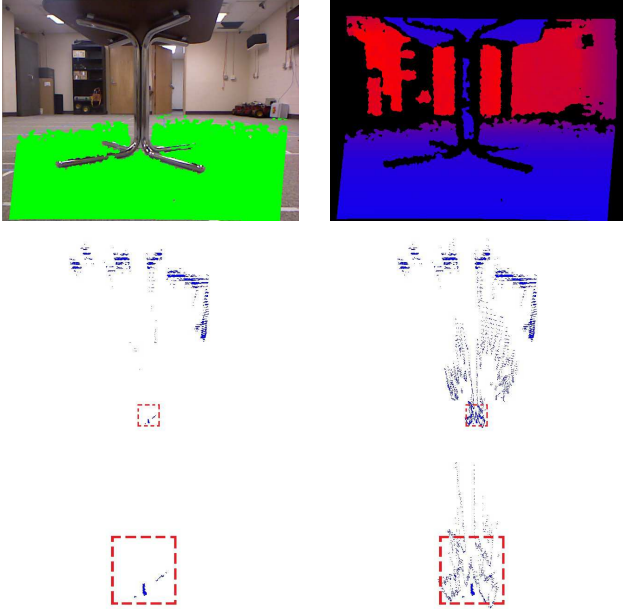


Fig. 3. TOP: An RGB image (left), with the segmented floor pixels colored green, and a depth image (right). No depth readings are available from the table support because of its reflective properties. MIDDLE: Occupancy map ignoring invalid readings (left) and using our infinite pole approach to handle specular surfaces (right). The red rectangle denotes the location of the table in the occupancy map. Note that the table top is intentionally not captured in the occupancy map because it is high enough for the robot to safely pass under it. BOTTOM: Zoomed-in version of the occupancy maps centered at the table’s location.

We call our technique the *infinite pole approach*. For each pixel in the depth image with an invalid depth reading, all adjacent pixels (using an 8-neighborhood) are examined. If any of these neighbors is a floor point, then a thin, vertical, infinitely tall obstacle is hypothesized at that floor point. On the other hand, if none of the neighbors is a floor point, then the invalid reading can safely be ignored because it will already be protected by a hypothesized infinite pole arising from a pixel below it in the depth map (assuming that the sensor is oriented in the usual manner). This approach assumes that specular obstacles rest on the floor, or at least rest on something that rests on the floor, causing it to fail only if a specular obstacle is hanging in the air with no direct connection to the ground. Figure 3 shows that this approach can successfully detect an object with a specular surface, despite not being able to obtain any depth readings of the object from the sensor. The top of the table in the figure is not captured in the occupancy map because it is high enough for the robot to safely pass under it. In this case, it is crucial that the system correctly handle the reflective material, because avoiding collision relies completely on detecting the reflective legs.

Another limitation of the sensor is its inability to produce depth readings if an obstacle is too close. Yet this is precisely the most dangerous situation for an obstacle avoidance system! The problem is caused by the projected IR pattern being

out of the field of view of the IR camera, thus preventing triangulation. Although our robot is designed to turn away from an object before it gets this close, the issue can arise in either the case of a dynamic obstacle that suddenly appears in front of the sensor, or when avoiding one obstacle the robot turns to face another nearby obstacle. To avoid collisions in either case, we implement a conservative strategy that takes advantage of the fact that the sensor yields invalid, rather than erroneous readings. If at any time more than $\tau_v = 0.4$ (i.e., 40%) of the pixels are found to have invalid readings, then the robot assumes that it is not safe to proceed further. In such a case the robot stops, then turns continuously until the condition no longer holds before it resumes driving. A plot of the percentage of non-depth pixels can be seen in Figure 4. One scenario that may arise is an object that is too close to the sensor for valid depth readings but too small to trigger the 40% test. In such an instance the infinite pole approach allows for the detection of the obstacle by marking, in the occupancy map, all invalid depth readings that are adjacent to floor pixels in the color image.

D. Obstacle Avoidance

The final step is to determine the robot’s translational and rotational velocities from the 2D occupancy grid. To make this decision, our approach is limited to examining a fixed rectangular region containing locations $x\rho_x \in [-0.25, 0.25]$ m and $y\rho_y \in [0.0, 1.0]$ m, that is, up to one meter in front of the robot and a quarter of a meter on either side of the center. Extensive work has been done by a number of authors on developing efficient and robust ways to avoid obstacles [12], [22], [7], [8], but we adopt a simpler approach based on that of Lorigo et al. [15]. The translational velocity is determined as a constant c_1 multiplied by the minimum distance to an occupied cell in the rectangular region of the 2D occupancy grid. The rotational velocity is calculated as the angle to the centroid of the occupied cells in the rectangular region, scaled by a constant c_2 :

$$v = c_1 \rho_y \arg \min_y \min_x m(x, y) \quad (8)$$

$$\omega = c_2 \arctan \mu_y / \mu_x, \quad (9)$$

where the centroid is given by

$$(\mu_x, \mu_y) = \frac{1}{\sum_{x,y} m(x, y)} \left(\sum_{x,y} x m(x, y), \sum_{x,y} y m(x, y) \right), \quad (10)$$

and $m(x, y)$ is the value of a cell in the occupancy map. As in [15], we also include a single bit of state to force the robot, once it begins rotating, to continue rotating in the same direction until the path is clear. This additional logic prevents the robot from getting stuck in a corner or tight space.

V. EXPERIMENTS

The robot used in our experiments was an ActivMedia P3AT Pioneer mobile robotics platform (approximately 0.5³ meters in size) with a computer mounted on top. The

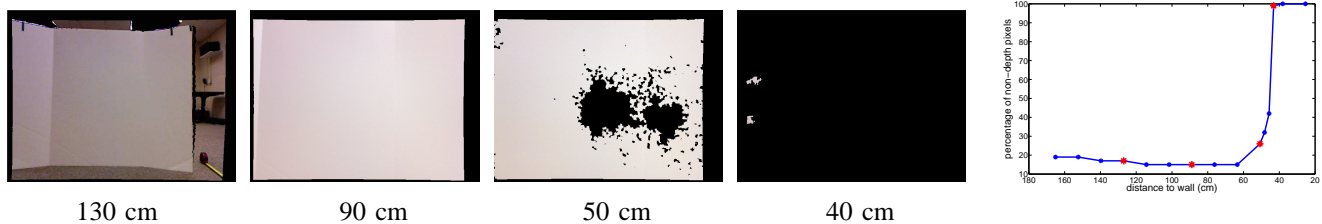


Fig. 4. LEFT: Selected images from a sequence obtained as the robot approached a wall, with the number under each image indicating the distance to the wall. Invalid depth readings are shown in black. RIGHT: Plot showing the percentage of invalid depth readings pixels as the robot approached the wall, with red asterisks corresponding to the images shown.

computer consisted of standard hardware with 6 GB RAM, an EVGA GeForce 9800 graphics card, and an Intel Core i7 processor running Ubuntu Linux 11.04. The robot was tethered through a network cable to a remote desktop for ease of interfacing. The 3D sensor was an Xbox Kinect 360 capable of providing RGB images, depth maps, and 3D point clouds at 30 frames per second, which is the same frame rate at which our system can detect obstacles.

To test the robustness of the obstacle detection and avoidance approach, the robot was placed in several environments containing various types of obstacles and lighting conditions. The lighting conditions included naturally lit rooms, artificially lit rooms, and rooms with no light. The types of obstacles used were dynamic obstacles, thin obstacles, and obstacles at multiple heights above the ground. The environments used for testing were as follows:

- 1) An empty room with several people walking around. The people were instructed to occasionally walk in front of the robot to enter the sensor's blind spot. This tested the system's ability to detect and avoid objects in a dynamic environment, as well as its ability to handle obstacles that enter the blind spot.
- 2) A room with several cables, as small as 0.5 cm in diameter, hanging from the ceiling. This tested the sensor's ability to detect thin objects.
- 3) A room with several bridges of varying height. This tested the system's ability to distinguish between obstacles under which the robot can safely pass, and those with which the robot will collide if attempting to pass under them.
- 4) A room with an object containing specular surfaces. This test the system's ability to handle obstacles that do not yield depth readings.

All these environments were tested with the three different lighting conditions.

Images of these environments are shown in Figure 5. At the start of each experiment, the robot was placed such that there were no obstacles directly in front of it, so that the floor plane could be estimated as described in Section IV-A. The floor plane calculated in this initial step was used for the rest of the experiment. The robot was given a start command through the network connection, after which it wandered around the environment autonomously, detecting and avoiding any obstacles that it encountered. The robot

ran until it received a stop command by a human operator.

In the environment with dynamic objects, the robot was able to successfully react to obstacles that moved into its path of motion and avoid them. The robot was also able to negotiate all the thin cables hanging from the ceiling in the second environment. In the third environment, the robot was able to successfully distinguish between objects that it could pass under and those that it could not. This allowed for the robot to achieve behavior impossible with most previous sensors, namely to distinguish between obstacles based on their height above the ground. The robot was also able to successfully avoid obstacles with specular materials.

In addition to these environments with the varying lighting conditions, the robot also wandered a hallway with a reflective tile floor typically found in public buildings and offices. This experiment revealed the robustness of the sensor to provide depth readings even for highly reflective surfaces.

We conducted an additional experiment to test the system's ability to detect and avoid obstacles at a high rate of speed. This is in the spirit of other work done by various researchers [16], [3], [10]. Our goal was to detect the frame rate of the sensor itself, the speed of the algorithm, and the ability of the overall system to robustly detect obstacles even when it was given little time to react. In this test, we ran the same detection and obstacle avoidance routines as the other experiments but with increased translation velocity from 0.2 m/s to the robot's maximum speed of 0.8 m/s and increased rotation velocities from 2.8 rad/s to 5.6 rad/s. Because our approach is able to update the occupancy map at 30 Hz, detecting and avoiding obstacles while the robot moves at high speed proved not to be an issue. All obstacles were avoided as they had been at slower speeds.

VI. CONCLUSION

In this paper we proposed an algorithm to detect and avoid obstacles for a mobile robot platform using a 3D sensor. Instead of projecting the depth map to UV-disparity, as performed by previous researchers, we adopt the strategy of projecting 3D points to the ground plane, from which a 2D occupancy map is filled. A simple control scheme produces translation and rotation velocities from the occupancy map. Our system utilizes a Kinect sensor, which facilitates real-time, accurate 3D measurements in an inexpensive manner. We have identified two limitations of the Kinect sensor, namely its inability to produce depth readings for specular

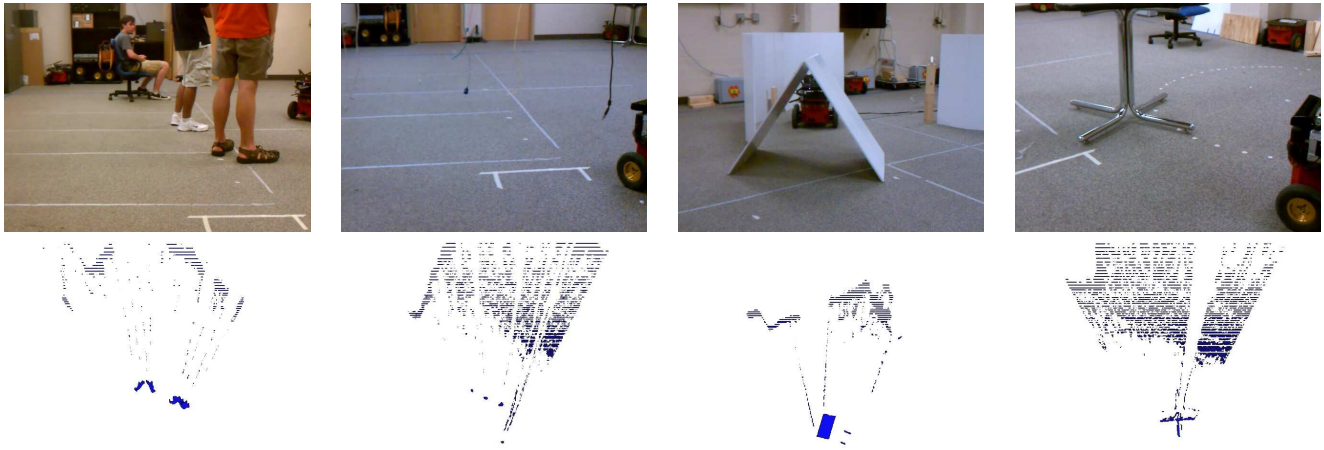


Fig. 5. TOP: Images of the various environments in which the obstacle detection and avoidance algorithm was run. From left to right: Dynamic environment, environment with thin objects, environment with multi-levelled obstacles, and an environment with reflective materials. BOTTOM: Occupancy maps of the environment as seen from the Kinect at the time of the top images.

surfaces and for objects that are too close to the sensor. We have proposed solutions to both of these problems. An infinite pole solution hypothesizes infinitely tall, thin obstacles when invalid depth readings are adjacent to ground plane pixels, and the control scheme turns away from scenes with a large percentage of invalid readings. Together, these innovations facilitate robust, real-time, reliable obstacle avoidance.

Our experiments demonstrate the feasibility of using a 3D sensor for obstacle detection and avoidance. The sensor contains several advantages over previous sensors, such as low power, low cost, high resolution, high frame rate, and the ability to provide readings outside of a horizontal plane. This last capability is particularly important, as it enables the robot to detect obstacles based on their height. As a result, as demonstrated by our system, the robot can determine whether or not it is low enough to drive under any particular obstacle. The sensor also has the advantage that it is unaffected by indoor lighting conditions, so that it performs equally well when all the lights are turned off. Future work will be aimed at avoiding obstacles such as ledges and stairs to prevent the robot from driving off ledges or falling into holes.

REFERENCES

- [1] M. Bai, Y. Zhuang, and W. Wang. Stereovision based obstacle detection approach for mobile robot navigation. In *International Conference on Intelligent Control and Information Processing (ICICIP)*, Aug. 2010.
- [2] P. Bouffard, J. Gillula, H. Huang, M. Vitus, and C. Tomlin. Ros code for obstacle avoidance using a quadrotor.
- [3] O. Brock and O. Khatib. High-speed navigation using the global dynamic window approach. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.
- [4] R. Brooks. A robust layered control system for a mobile robot. *IEEE Transactions on Robotics and Automation*, 2(1):14–23, Mar. 1986.
- [5] J. Cunha, E. Pedrosa, C. Cruz, A. J. R. Neves, and N. Lau. Using a depth camera for indoor robot localization and navigation. In *Robotics Science and Systems (RSS) RGB-D Workshop*, June 2011.
- [6] F. Flacco, T. Kröger, A. D. Luca, and O. Khatib. A depth space approach to human-robot collision avoidance. In *Proceedings of the International Conference on Robotics and Automation*, May 2012.
- [7] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1), 1997.
- [8] D. Fox, W. Burgard, S. Thrun, and A. B. Cremers. A hybrid collision avoidance method for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1998.
- [9] Y. Gao, X. Ai, J. Rarity, and N. Dahnoun. Obstacle detection with 3D camera using U-V-disparity. *Internal Workshop on Systems, Signal Processing and their Applications (WOSSPA)*, 2011.
- [10] N. Hany, M. Hanif, and T. Yaw. Modeling of high-speed vehicle for path following and obstacles avoidance. *The 18th IASTD International Conference on Modeling and Simulation*, 2007.
- [11] Z. Hu, F. Lamosa, and K. Uchimura. A complete U-V-disparity study for stereovision based 3D driving environment analysis. *3-D Digital Imaging and Modeling (3DIM)*, 2005.
- [12] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research (IJRR)*, 5(1):90–98, Mar. 1986.
- [13] J. Kim and Y. Suga. An omnidirectional vision-based moving obstacle detection in mobile robot. *International Journal of Control, Automation, and Systems (IJCAS)*, 5(6):663–673, Dec. 2007.
- [14] M. Kumano, A. Ohya, and S. Yuta. Obstacle avoidance of autonomous mobile robot using stereo vision sensor. In *Intl. Symp. Robot. Automat.*, pages 497–502, 2000.
- [15] L. M. Lorigo, R. A. Brooks, and W. E. L. Grimson. Visually-guided obstacle avoidance in unstructured environments. In *IEEE Conference on Intelligent Robots and Systems*, volume 1, pages 373–379, Sept. 1997.
- [16] J. Michels, A. Saxena, and A. Y. Ng. High-speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the Twenty-second International Conference on Machine Learning (ICML)*, pages 593–600, 2005.
- [17] R. Mojtahedzadeh. Robot obstacle avoidance using the Kinect. Master’s thesis, KTH Computer Science and Communication, 2011.
- [18] R. C. Nelson and J. Aloimonos. Using flow field divergence for obstacle avoidance towards qualitative vision. In *Proceedings of the International Conference on Computer Vision*, pages 188–196, 1988.
- [19] I. Nourbakhsh. The sonars of Dervish. *The Robot Practitioner*, 1(4), 1995.
- [20] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.
- [21] S. Soumare, A. Ohya, and S. Yuta. Real-time obstacle avoidance by an autonomous mobile robot using an active vision sensor and a vertically emitted laser slit. In *Intelligent Autonomous Systems*, pages 301–308, 2002.
- [22] I. Ulrich and J. Borenstein. VFH+: Reliable obstacle avoidance for fast mobile robots. *International Conference on Robotics and Automation (ICRA)*, pages 1572–1577, May 1998.