Ground Plane Detection Using an RGB-D Sensor

Doğan Kırcalı and F. Boray Tek

Robotics and Autonomous Vehicles Laboratory,
Computer Engineering Department, Işık University, 34980, Şile, İstanbul, Turkey
{dogan,boray}@isikun.edu.tr
http://ravlab.isikun.edu.tr

Abstract. Ground plane detection is essential for successful navigation of vision based mobile robots. We introduce a very simple but robust ground plane detection method based on depth information obtained using an RGB-Depth sensor. We present two different variations of the method: the simplest works robust for setups where the sensor pitch angle is fixed and has no roll, whereas a second version can handle changes in pitch and roll angles. The comparative experiments show that our approach performs better than the vertical disparity approach. It produces acceptable and useful ground plane-obstacle segmentations for many difficult scenes which include many obstacles, different floor surfaces, stairs, and narrow corridors.

Keywords: Ground Plane Detection, Kinect, Depth-Map, RGB-D, Autonomous Robot Navigation, Obstacle Detection, V-Disparity.

1 Introduction

Ground plane detection and obstacle detection are essential tasks to determine passable regions for autonomous navigation. To detect the ground plane in a scene the most common approach is to utilize depth information (i.e. depth map). Various methods and sensors have been used to compute the depth map of the scene.

Recent introduction of RGB-D sensors (Red-Green-Blue-Depth) allowed affordable and easy computation of depth maps. Microsoft Kinect is a pioneer of such sensors which was initially marketed as a peripheral input device for computer games. It integrates an infrared (IR) projector, a RGB camera, a monochrome IR camera, a tilt motor and a microphone array. The device can be used to obtain 640x480 pixel depth map and RGB video stream at a rate of 30fps.

Kinect uses an IR laser projector to cast a structured light pattern to the scene. Simultaneously, an image of the scene is acquired by a monochrome CMOS camera. The disparities between the expected and the observed patterns are used to estimate a depth value for each pixel. Kinect works quite well for indoor environments. However, the depth reading is not reliable for regions that are far

more than 4 meters; at the boundaries of the objects because of the shadowing; reflective or IR absorbing surfaces; and at the places that are illuminated directly by sunlight which causes IR interference. Accuracy under different conditions was studied in [1–3].

Regardless of the method or the device that is used to obtain depth information there are several works which approach to the ground plane detection problem based on the relationship between a pixel's position and it is disparity[4–9]

Li *et al.* show that the vertical position (y) of a pixel of the ground plane is linearly related to its disparity $D(y)$ such that one can seek a linear equation $D(y) = K1 + K2 * y$, where $K1$ and $K2$ are constants which are determined by the sensor's intrinsic parameters, height, and tilt angle. However, ground plane can be directly estimated on the image coordinates using the plane equation based on disparity $D(x, y) = ax + by + c$ without determining mentioned parameters. A least squares estimation of the ground plane can be performed offline (i.e. by pre-calibration) if a ground plane only depth image of the scene is available [5]. Another common approach is to use RANSAC algorithm which allows fitting of the ground plane even the image includes other planes [10, 11, 4]. Since RANSAC is used to estimate linear planes, the ground plane is assumed to be the dominant plane in the image.

There are some other works of segmentation of the scene into relevant planes [12, 11]. The work of Holz *et al.* clusters surface normals to segment planes and reported to be accurate in close ranges[11].

In[7] row histograms of the disparity image are used to model the ground plane. In the image formed of the row histograms (named as *V-disparity*), the ground plane appears as a diagonal line. This line, which is detected by Hough Transform, was used as the ground plane model.

In this paper, we present a novel and simple algorithm to detect the ground plane without the assumption of that it is the largest region. Our method is based on the fact that if a pixel is from the ground plane, its depth value must be on a rationally increasing curve placed on its vertical position. However, the degree of this rational function is not fixed due to reasons which we explain later. Nevertheless, it can be easily estimated by an exponential curve fit which can be used as a ground plane model. Later, the pixels which are consistent with the model are detected as ground plane whereas the others are marked as obstacles. While this is our base model which can be used for a fixed viewing angle scenario, we provide an extension of it for dynamic environments where sensor viewing angle changes from frame to frame. Moreover, we note the relation of our approach to the V-disparity approach[7], which rely on the linear increase of disparity and fitting of a linear line to model the ground plane. Thus, we provide experiments which test and compare both approaches on the same data.

This paper is organized as follows: In Section 2 we present the proposed method. Section 3 presents the results of the experiments. Our conclusion and future work are presented in Section 4.

2 Method

2.1 Detection for Fixed Pitch

In a common scenario, the sensor views the ground plane with an angle (i.e. pitch angle). The sensor's pitch angle (Figure 1(a)) causes allocation of more pixels for the closer locations of the scene than the farther parts. So that linear distance from the sensor is projected on the depth map as a rational function. This is demonstrated by an example of the intensity coded depth map image obtained from Kinect (Figure 1(c)). Any column of the depth image will show that the depth value increases not linearly but exponentially from bottom to top (i.e. right to left in Figure 1(d)).

In this section we assume that the sensor is fixed and its roll angle is zero (Figure 1(b)). Furthermore, a "ground plane only" depth image will have all columns equal to each other. These columns are estimable by an exponential function.

Thus, we can fit a curve to any vertical line of the depth map. We found that a good fit is possible with sum of two exponential functions in the following form:

$$f(x) = ae^{bx} + ce^{dx} \tag{1}$$

where $f(x)$ is the pixel's depth value and x is the its vertical location (i.e. row index) in the image. The coefficients (a, b, c, d) depend on the intrinsic parameters, pitch angle, and the height of the sensor.

These coefficients are estimated by a least squares fitting method. Then it is possible to reconstruct a curve, which we call as the *reference ground plane curve* (C_R).

In order to detect ground plane pixels in a new depth map, the columns of the new depth map (C_U) are compared to C_R. Any value that is under C_R represents an object (or any protrusion), whereas values above the reference curve represent drop-offs, holes (e.g. intrusions, downstairs, edge of a table) in the scene. Hence we compare the absolute difference against a pre-defined threshold value T; mark the pixels as ground plane if difference is less than T.

For the comparison, depth values that are zero, ignored as they indicate sensor reading errors. The experiments concerning this part are presented in Section 3.

2.2 Detection for Changing Pitch and Roll

The fixed pitch angle scheme explained above is quite robust. However, it is not suitable for the scenarios where the pitch and roll angles of the sensor changes. Generally the mobile robots exhibit movements on the sensors' platform. Pitch and roll movements can be compensated by using an additional gyroscopic stabilization [13]. However, here we propose a computational solution. In this approach we do not calculate a reference ground curve from a reference pre-calibration image but estimate it each time from the particular input frame.

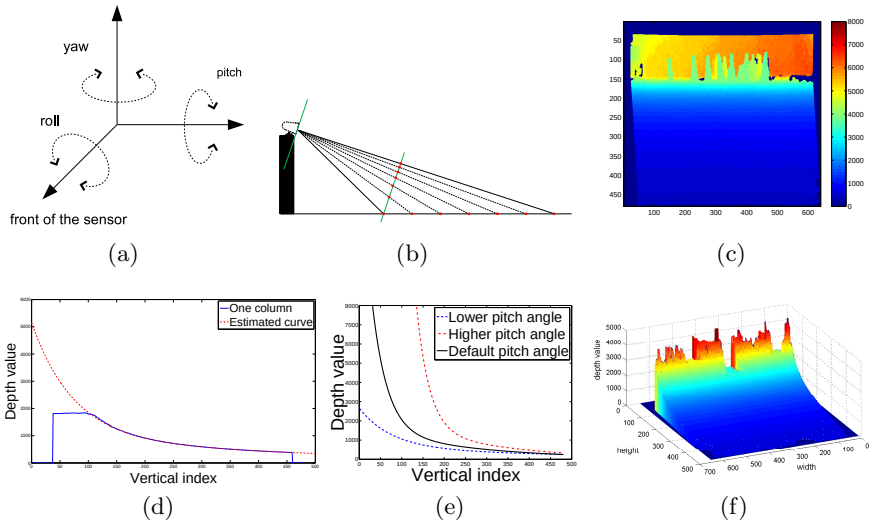

Fig. 1. (a) Roll & pitch axis, (b) sensor view pitch causes linearly spaced points to mapped as an exponential increasing function.(c) An example depth map image, (d) one column (y=517) of the depth map and its fitted curve representing the ground plane, (e) ground plane curves for different pitch angles, (f) depth map in three dimensions showing the drop-offs caused by the objects.

A higher pitch angle (sensor almost parallel to the ground) will increase the slope of the ground plane curve. Whereas a non-zero roll angle (horizontal angular change) of the sensor forms different ground plane curves along columns of the depth map (Figure 1(e)). Such that at one end the depth map exhibits curves of higher pitch angles while towards the other end having curves of lower pitch angles. These variations complicate the use of a single reference curve for that frame.

To overcome roll angle affects our approach aims to rotate the depth map to make it orthogonal to the ground plane. If the sensor is orthogonal to the ground plane it is expected to produce equal or very similar depth values along every horizontal line (i.e. rows). And this similarity can be simply captured by calculating a histogram of the row values such that a higher histogram peak value indicates more similar values along a row. Let h_r shows the histogram of the rth row of a depth image (D) of R rows, and let us denote the rotation of depth image with D_θ.

$$argmax_\theta(\sum_{r=1}^{R} argmax_i(h_r(i, D_\theta)) \tag{2}$$

Thus for each angle value θ in a predefined set, the depth map is rotated with an angle θ and the histogram h_r is computed for every row r. Then, the angle θ that gives the total maximum peak histogram value (summed over rows)

is estimated as the best rotation angle. This angle is used to rotate the depth map prior to the ground plane curve estimation. After the roll affect is removed the pitch compensation curve estimation scheme can start.

As explained, changes of pitch angle create different projection and different curves (Figure 1(e)). Moreover, since the scene may contain obstacles we must define a new approach for ground plane curve estimation.

In a scene that consists of both the ground plane and objects, as in Figure 1(f), maximum value along a particular row of the depth map must be due the ground plane, unless an object is covering the whole row. This is because the objects that are closer to the sensor than the ground plane surface that they occlude. Therefore, if the maximum value across each row (r) of the depth map (D) is taken, which we name as the *depth envelope* (E), it can be used to estimate the *reference ground plane curve* (C_R) for this particular depth frame.

$$E(r) = max_i(D(c_i, r)) \tag{3}$$

The estimation is again performed by fitting the aforementioned exponential curve (1). Prior to the curve fitting we perform median filtering to smooth the depth envelope. Moreover, depth values must increase exponentially from bottom of the scene to the top. However, when the scene ends with a wall or group of obstacles this is reflected as a plateau in the depth envelope. Hence the envelope (E) is scanned from right to left and the values after the highest peak are excluded from fitting as they cannot be a part of the ground plane.

There are two conditions which affect the ground plane curve fit adversely. First, when one or more objects cover an entire row, this will produce a plateau in the profile of the depth map. However, if the rows of the "entire row covering object or group" do not form the highest plateau in the image, ground plane continues afterwards curve continues and the object will not affect the curve estimation.

Second, any drop-offs exhibit higher depth values than the ground plane: drop-offs cause sudden increases (hills) on the depth envelope. If a hill is found on the depth envelope, the estimated curve will be produced by a higher fitting error.

After estimating the ground plane reference curve coefficients for the frame, every column is compared with the reference curve as it was done for Section 2.1. The pixels are classified as ground plane and non-ground plane by comparing against a threshold T. The value of T was determined by overall accuracy.

A point to note is about non-planar ground surfaces that few other studies in literature have devised strategies for [7, 6]. We assume here a planar ground plane model which will probably cause problems if the floor has bumps or significant inclination or declination [7]. Our future work will focus on these aspects.

3 Experiments

We run our algorithm on four different multi-frame data sets that were not used in the development phase. The dimensions of the depth map and RGB images are

640x480. Two of these datasets (dataset-1 and dataset-2) were manually labeled to provide ground truth and were used in plotting ROC (Receiver Operating Curves), whereas the other two were manually (visually) examined. Dataset-1 and dataset-2 composed of 300 frames captured on a mobile robot platform which moves in the laboratory floor among obstacles. Dataset-3 created with the same platform; however, the pitch and roll angles change excessively. Dataset-4 included 12 individual frames acquired from difficult scenes such as narrow corridors, wall only scenes etc.

We compare three different versions for our approach: A1-fixed pitch, A2-pitch compensated, A3-pitch and roll compensated. There is only one free parameter for A1 and A2 that is threshold T, which is estimated by ROC analysis; whereas the 3rd roll compensation algorithm requires pre-defined angle set to search for best rotation angle: $\{-30°, -28°,..,+30°\}$. Least squares fit was performed by Matlab curve fitting function with default parameters. However, we excluded the depth values which are equal to zero, or above 5000 due to inaccurate sensor readings. Additionally, as explained previously, for algorithm A2 and A3 the indices positioned to left of the maximum of the column depth value must be excluded from the fits since they do not represent ground plane. Finally, note that A1 requires a onetime pre-calibration and estimation of the coefficients for the reference ground plane curve, whereas A2 and A3 estimate coefficients separately for each new frame.

Moreover, we compare the results with V-disp method [7]. We note that V-disp is originally developed for stereo depth calculation where disparity is available before depth. To implement V-disp method by Kinect depth stream, we calculated disparity from the depth map (i.e. $1/D$), calculated row histograms to form V-disp image, and then run Hough transform to estimate ground plane line. We had to put a constraint on the Hough line search in $[-60°, -30°]$ range to have relevant results.

Since A3 and A2 algorithms are same except for the roll compensation, we will examine and compare results of A2 to A1 and V-disp; however we compare A3 results only against A2 to show the effect of roll compensation scheme.

Figure 2(a) and 2(b) show ROC curves and overall accuracies plotted for our fixed and pitch compensated algorithms (A1 and A2) and V-disp method on dataset-2. It can be seen that our pitch compensated algorithm is superior to both V-disp which is better than our fixed algorithm.

When we select best accuracy point thresholds and run our algorithms on dataset-2, we are able to see accuracy vs. frames (Figure 2(c)). In addition we record curve fitting error for pitch compensated algorithm (A2). It can be seen that both methods are quite stable with the exception being high curve fitting error frames for A2. It is also easy to spot these frames on live data sequences.

Beside multi-frame datasets, we included here some example single input-output pairs (Figure 3). Here ground plane is marked with black and obstacles were marked with white to ease viewing. In Figure 3(a), we observe a cluttered scene. Note that its depth map contained sensor reading errors because of the lighting and reflective patches (Figure 3(b)). The output of A2 is shown in right

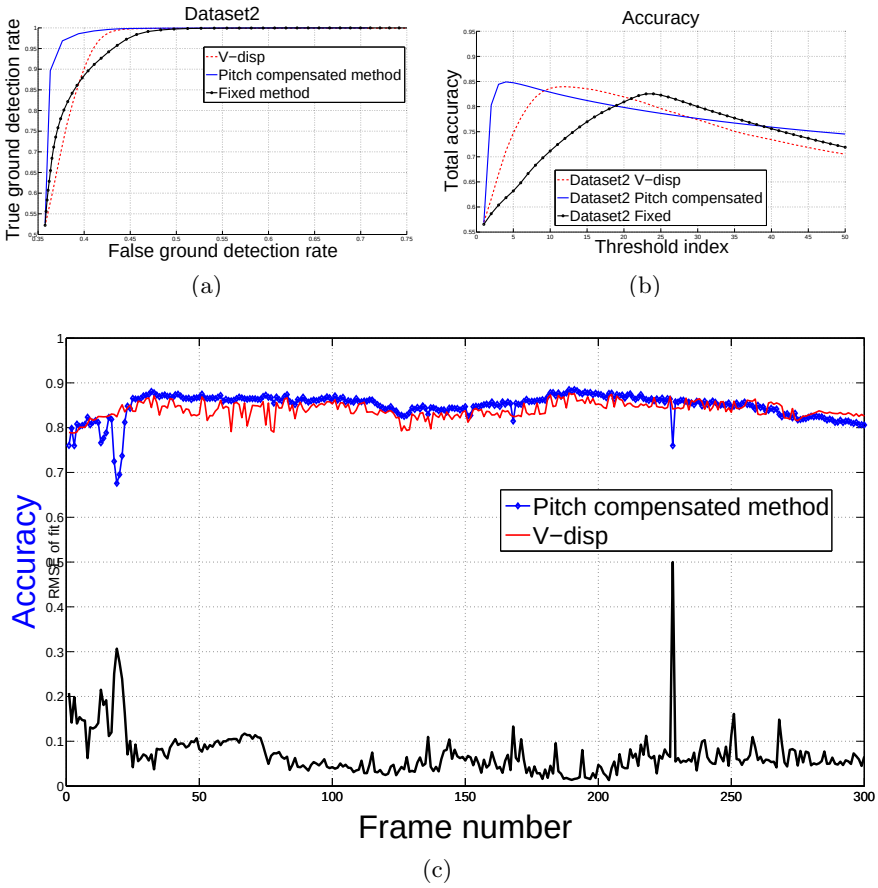

Fig. 2. a) ROC curves comparing V-disp and our fixed and pitch compensated algorithms (A1-A2), b) average accuracy over 300 frames vs. thresholds, c) accuracy and curve fit error of A2 for individual frames.

column (Figure 3(c)). It can be seen that algorithm is quite successful in the regions where there is depth reading. Despite that it is possible to reduce the spurious noisy detections; we show here the raw outputs.

Figure 3(d),3(e),3(f)) show another difficult scene where the robot with sensor is positioned in front of stairs. Due to reflective marble floor the sensor produce many zeros in the close ground plane. In addition, we observe many zeros in distant walls. However, the output is quite successful in the sense that the close plan ground floor and the edge of the stairs is correctly identified.

Despite that dataset-1 and 2 are similar, dataset-3 contains excessive roll changes which were used to test roll compensation (A2 vs. A3). The outputs show that roll compensation is able to detect and correct rotations. Figure 3(g)) show one of the frames from dataset-3, where the sensor is rolled almost 20°

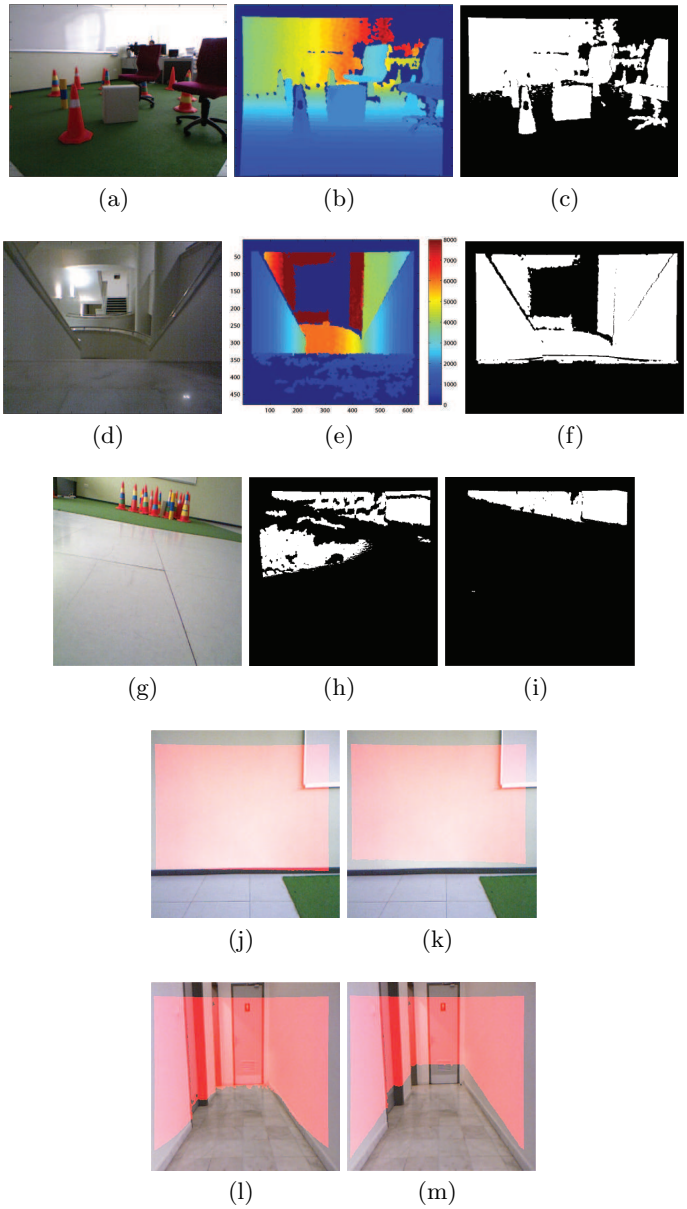

Fig. 3. Experimental results from different scenes. RGB, depth-map and pitch compensated method output (white pixels represent objects whereas black pixels represent ground plane): (a,b,c) lab environment with many objects and reflections; (d,e,f) stairs (g,h,i) respective outputs of pitch compensated (A-2) and pitch&roll compensated method on an image where sensor was positioned with a roll angle (A-3). Comparison of pitch compensated (left) and V-disp method (right) in different scenes: (j,k) wall surface as dominant plane, (l,m) narrow corridor.

degrees. Figure 3(h),3(i) shows the respective outputs of A2 and A3. It can be seen that roll compensation provides a significant advantage if sensor can roll.

Finally, Figure 3(l)-3(m) shows output pairs (overlayed on RGB) for A2 and V-disp. It can be seen that both methods can detect ground planes in scenes where ground plane is not the largest or dominant plane. Both methods thresholds are fixed as they produce the highest respective overall accuracies in datasets 1 and 2. Note that V-disp marked more non-passable regions as ground plane.

If the frames are buffered beforehand and worked offline, our pitch compensated algorithm A2 processed 83 fps while running on a computer with Pentium i5 480m processor using Matlab 2011a.

Additional experimental results and datasets can be found from our web site[1].

4 Conclusion

We have presented a novel, and robust ground plane detection algorithm which uses depth information obtained from an RGB-D sensor. Our approach includes two different methods, where the first one is simple but quite robust for fixed pitch and no-roll angle scenarios, whereas the second one is more suitable for dynamic environments. Both algorithms are based on an exponential curve fit to model the ground plane which exhibits rational decreasing depth values. We compared our method to the popular V-disp [7] method which is based on detection of a ground plane model line by Hough transform which relied on linear increasing disparity values.

We have shown that the proposed method is better than V-disp and produces acceptable and useful ground plane-obstacle segmentations for many difficult scenes, which included many obstacles, different surfaces, stairs, and narrow corridors.

Our method can produce erroneous detections especially when the curve fitting is not successful. However, these situations are easy to detect by checking the RMS error of the fit which has been shown to be highly correlated with the accuracy of segmentation. Our future work will include an iterative refining procedure for curve fitting for the frames which are detected to produce high RMS fitting errors.

Acknowledgments. All necessary components in this paper were financed by FMV Işık University internal research funds BAP-10B302 project.

[1] `http://ravlab.isikun.edu.tr`

References

1. J. Stowers, M. Hayes, and A. Bainbridge-Smith. Altitude control of a quadrotor helicopter using depth map from microsoft kinect sensor. In *Mechatronics (ICM), 2011 IEEE Int. Conference on*, pages 358–362, April.
2. Caroline Rougier, Edouard Auvinet, Jacqueline Rousseau, Max Mignotte, and Jean Meunier. Fall detection from depth map video sequences. In *ICOST'11*, pages 121–128, Berlin, Heidelberg, 2011.
3. Kourosh Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2), 2012.
4. F Li, J M Brady, I Reid, and H Hu. Parallel image processing for object tracking using disparity information. In *In Second Asian Conference on Computer Vision ACCV '95*, pages 762–766.
5. Stephen Se and Michael Brady. Ground plane estimation, error analysis and applications. *Robotics and Autonomous Systems*, 39(2):59 – 71, 2002.
6. Qian Yu, Helder Araújo, and Hong Wang. A stereovision method for obstacle detection and tracking in non-flat urban environments. *Auton. Robots*, 19(2):141–157, September 2005.
7. R. Labayrade, D. Aubert, and J. P Tarel. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 2, pages 646–651 vol.2, June.
8. Camillo J. Taylor and Anthony Cowley. Parsing indoor scenes using rgb-d imagery. In *Robotics: Science and Systems*, July 2012.
9. K. Gong and R. Green. Ground-plane detection using stereo depth values for wheelchair guidance. In *Image and Vision Computing New Zealand, 2009. IVCNZ '09.*, pages 97–101.
10. C. Zheng and R. Green. Feature recognition and obstacle detection for drive assistance in indoor environments. In *Image and Vision Computing New Zealand, 2011. IVCNZ '11.*
11. Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-Time Plane Segmentation using RGB-D Cameras. In *Proceedings of the 15th RoboCup Int. Symposium*, volume 7416, pages 307–317, Istanbul, Turkey, July 2011. Springer.
12. Can Erdogan, Manohar Paluri, and Frank Dellaert. Planar segmentation of rgbd images using fast linear fitting and markov chain monte carlo. In *CRV'12*, pages 32–39, 2012.
13. Luke Wang, Russel Vanderhout, and Tim Shi. Computer vision detection of negative obstacles with the microsoft kinect. *University of British Columbia. Engineering Projects Project Lab. ENPH 459, Project Conclusion Reports*, 2012.