Real-Time Plane Detection Based on Depth Map from Kinect

Hyun Woo Yoo, Woo Hyun Kim, Jeong Woo Park, Won Hyong Lee, and Myung Jin Chung
Department of Electrical Engineering, KAIST, Daejeon, 350-701, Korea
(Tel: 82-42-350-3429, E-mail: {yulgokee, ishsrain, pjw, leestation}@rr.kaist.ac.kr, mjchung@ee.kaist.ac.kr)

***Abstract*-- This paper suggests a system to detect multiple planes fast and roughly based on a depth map from depth camera 'Kinect'. In order to improve the speed of the plane detection process, we compute local normal vectors of points of the depth map and we classify 3D point cloud data whether it includes in the same plane using these local normal vectors. It is possible to detect multiple planes simultaneously. Experimental results show that the proposed method is faster than previous plane detection methods such as 3D Hough Transform and RANSAC, and also works in real-time.**

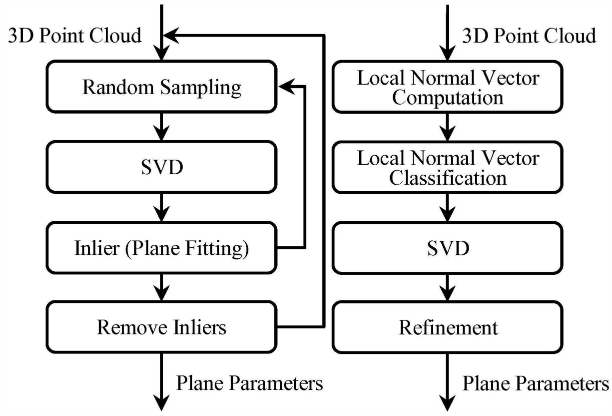
Index Terms—**Plane Detection, Depth Map, Kinect, Real-Time**

I. Introduction

Recently, many automatic robots have been developed for the work of humans in factories and offices. Robots have to detect the 3D model or position of their environment like a worktable, a workspace, or objects. In this paper, we assume the environment is a manufacturing process of a ship. It has characteristics which is sealed, lightless, and right-angled. To detect planes in this sealed and lightless environment, we use a Microsoft Kinect camera [1]. It has an infrared sensor. Therefore, it is possible to get a depth map in sealed and lightless environment. Also, it provides both a color image stream and a depth map stream in real-time. Therefore, we can get the 3D point cloud data from Kinect easily. On the depth map, we can compute local normal vectors of 3D point cloud data fast instead of computing 3D distance of each 3D point cloud data. Then we classify the data by whether it is included in a same plane using its local normal vectors computed.

In previous plane detection methods [2][3], problems related with processing speed have been occurred. The 3D Hough Transform [2] uses all points of 3D point cloud, it has high computational cost. The RANSAC [3] is unlimited to number of iteration. Thus, processing time of RANSAC is cannot be guaranteed. The block diagram of RANSAC is shown as Fig. 1 (a). This causes the disadvantage of the execution time. If we take reliable points instead of random points, we reduce the iteration process.

In this paper, inspired by this, we propose a real-time plane detection method based on depth map from Kinect. The block diagram of the proposed method is shown as Fig. 1 (b). This paper is organized as follows. We present the proposed method in the Section 2, experimental result in the Section 3, and conclusion in the Section 4.

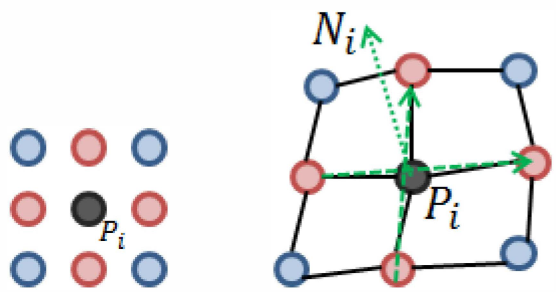


(a) (b)
Fig. 1. Block diagram of (a) RANSAC, (b) the proposed method

II. The Proposed Method

A. Computation of Local Normal Vectors

By using depth map, we can compute local normal vectors of 3D point cloud data very simply. Assuming that four nearest point : up, down, left, and right around the data point on a depth map are to be computed for a local normal vector of 3D point cloud. The process of computing 3D distance of each 3D point cloud data is simplified.



(a) Points of the Depth Map (b) Points of 3D point cloud
Fig. 2. Computation of Local Normal Vectors

The local normal vector computation method by using depth map is follow as. Compute the local normal vector of point P_i, select the four points on the depth map up, down, left, and right around the point P_i (Fig. 2 (a)), set the vector between up-down, and left-right in the 3D point cloud (Fig. 2 (b)), and cross product them. Finally we get a local normal vector N_i of point P_i.

B. Classification of 3D Point Cloud Data

Instead of computing local normal vector of every 3D point cloud data, the data are uniformly sampled on the depth map for increasing speed of process. As local normal vectors of uniformly sampled data points are computed, each of points is checked whether it is included in the same plane. Conditions that two different points are located in the same plane are as the following Equation (1) and (2).

$$N_i \cdot N_j = 1 \tag{1}$$
$$(P_i - P_j) \cdot N_i = 0 \tag{2}$$

Where, N_i and N_j are local normal vector of point P_i and P_j. Equation (1) checks that local normal vectors of each point are same. Equation (2) checks that a line formed by two points and local normal vector are perpendicular. Then, the points are classified as each other with in the same plane.

C. Plane Detection form Classified 3D Point Cloud Data

After classifying the 3D point cloud, the plane equation of each classified 3D point cloud data is found. A plane is described as an equation in the form Equation (3). In order to detect a plane from multiple points, the following matrix as Equation (4) is configured.

$$ax+by+cz+d=0 \tag{3}$$
$$\begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = 0, \tag{4}$$
$$P_n = (x_n, y_n, z_n),\ P_1, \cdots, P_n \in X_i$$

In this equation, X_i is a point group which is i-th separated, n is number of points in the group. The solution of these over-determined systems is found by the using the SVD (Singular Value Decomposition) on least square method and the solution is a coefficient in the equation of the plane.

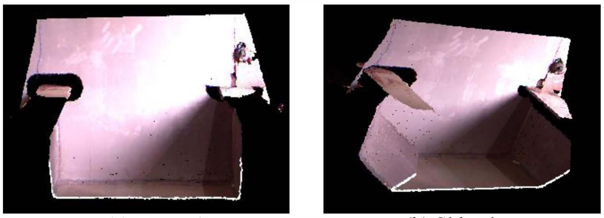
D. Similar Plane Discard through Comparison of Plane Equation Coefficients

Finally, similar planes are discarded by checking the similarity between other planes. Then the plane which has the most number of points contained is maintained and remaining planes are discarded. Conditions that two planes are same as the following Equation (5) and (6).

$$(a_i, b_i, c_i) \cdot (a_j, b_j, c_j) = 1 \tag{5}$$
$$d_i - d_j = 0 \tag{6}$$

In this equation, a, b, c, d are coefficient of plane equation of plane i and j. Equation (5) checks that the angles of two planes are same. If equation (5) is satisfied, equation (6) checks that the distance between two planes is zero. In this section, we set thresholds to recognize similar planes. If equation (5) and (6) are satisfied by these thresholds, the plane with a small number of points is removed by recognizing the similar plane.

III. EXPERIMENTAL RESULT



(a) Front view (b) Side view

Fig. 3. Obtained 3D point cloud data in experiment environments

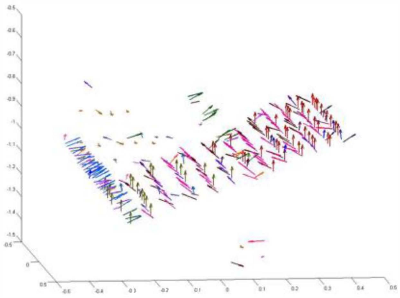


Fig. 4. Classified local normal vectors

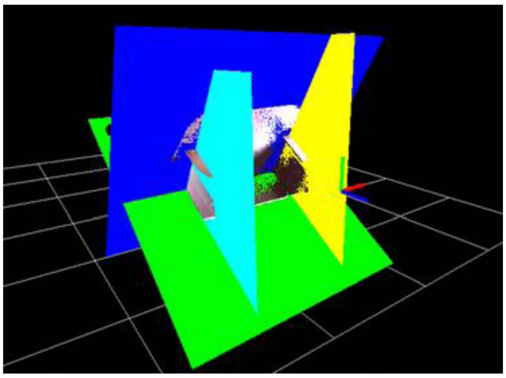


Fig. 5. Detected planes using the proposed method

The proposed method was tested on PC with Intel Core i5 2.80GHz processor. Fig. 3 shows the 3D point cloud data from the environment of the hull blocks in order to confirm the performance of the proposed method. There are roughly four planes (front, left, right, and floor) in Fig. 3.

Fig. 4 shows classified local normal vectors and Fig. 5 shows finally detected planes using the proposed method. In Fig. 4, classified point cloud can be found through colors of local normal vectors, and we can find that the data points are over classified because the number of classified point clouds is more than four that the expected number of planes. The exact number of point clouds is 30. After discarding similar planes, finally planes are detected as shown Fig. 5 which is similar with 3D point cloud data in Fig. 3.

TABLE I

COMPARISON OF PROCESS TIMES

	The Proposed Method		3D Hough Transform	RANSAC
Environment	C++	MATLAB		
Process Time	0.002 sec	0.522 sec	409 sec	6.74 sec

Table I shows a comparison of processing times in the

MATLAB environment. We can check that the processing speed of the proposed method is the fastest than other methods. In the proposed method, the data points are sampled on the 30 pixels each horizontal and vertical direction. In the 3D Hough transform, the resolution of parameter space is $\Delta\varphi$=1º, $\Delta\theta$=1º, $\Delta\rho$=0.01m. RANSAC is limited to 50 times to find the maximum number of iterations, and it is repeated the process that remove the point which is included the plane find out, and find another plane again until 1/100 of initial number of points of 3D point clouds. And the proposed method works in 0.002 second at C++ environment. This shows that the propose method works on real-time.

TABLE II
COMPARISON BETWEEN RESULTS OF THREE PLANE DETECTION METHODS

	The Proposed Method	3D Hough Transform	RANSAC
Angle between Front-Floor planes	89.2733º	89.9414º	90.1486º
Angle between Front-Left planes	90.1249º	91.1986º	91.0169º
Angle between Front-Right planes	91.0661º	89.8276º	90.1900º
Angle between Left-Right planes	178.8444º	178.0518º	178.5675º
Distance between Left-Right planes	0.8380m	0.8400m	0.8281m

Table II shows a comparison of results of the proposed method and other previous methods. We can check that angles or distance of detected planes are almost same. In real environment, the angle between planes is 90º. However, it can be seen that there are some errors in three methods. The distance of the wall between the left and right is 0.85m in a real environment, however there are some errors in three methods. The most accuracy method is 3D Hough transform, followed by the proposed method, and RANSAC.

TABLE III
COMPARISON OF ERRORS

	The Proposed Method	3D Hough Transform	RANSAC
Front plane	0.0025m	0.0111m	0.0028m
Floor plane	0.0023m	0.0111m	0.0019m
Left plane	0.0010m	0.0066m	0.0009m
Right plane	0.0021m	0.0123m	0.0021m

In order to quantitative evaluation, we picked reliable points at the whole 3D point cloud data and made a ground truth data. Table III shows errors between ground truth data and detected plane equations. Equation (7) shows how to calculate errors.

$$err_i = \frac{1}{N}\sum_{k=1}^{N} a_i x_k + b_i y_k + c_i z_k + d_i \qquad (7)$$

As a result, the performance of the 3D Hough transform is worst, the proposed method and the RANSAC method have error of 1-2mm levels. There is no significant difference between the two algorithms.

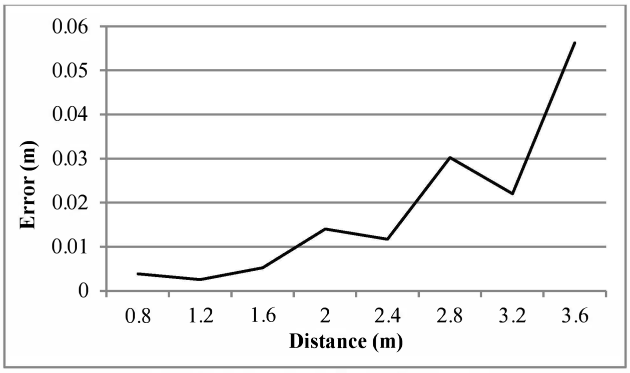
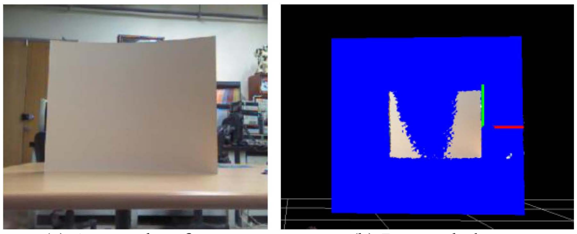


Fig. 6. Error corresponding to the distance

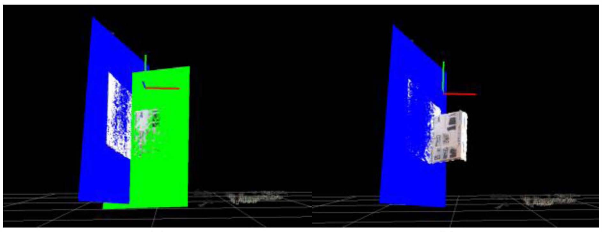
Fig. 6 shows error corresponding to the distance. The error was also calculated by equation (7), and it can be seen that the error increases as distance increases. As distance increases, depth map of the Kinect has vibration more. Because of this, the error seems to increase.

Also, the proposed method is suitable for right-angle environments, because it is based on the local normal vector. For this reason, planes cannot be detected correctly at a curved surface such as Fig. 7 or even slightly off vertically such as Fig. 8.



(a) A curved surface (b) Detected plane

Fig. 7. Failure case: a curved surface



(a) (b)

Fig. 8. Failure case: out of perpendicular planes. The angle between two planes is (a) 90 º, (b) 80 º

IV. CONCLUSIONS

In this paper, we have proposed a real-time plane detection method based on depth map from Kinect. The proposed method detects multiple planes fast by computing local normal vectors using characteristic of Kinect and using reliable normal vectors which is classified. As a result, the proposed method has execution time of 2ms and error of 1~2mm levels. In contrast, 3D Hough Transform method needs not only many execution times, but also many memory spaces to increase resolution of accumulator for increasing accuracy. Also the proposed method is not iterative, so the proposed method is faster than RANSAC. However the accuracy is almost same because using only reliable local normal vectors.

For further research, we can add computing process for more accuracy although execution time is longer. Also it is important to extract highly reliable local normal vectors in the proposed method. Therefore we have to study to reduce the noise of 3D point cloud and to improve classifying local normal vectors method.

ACKNOWLEDGMENT

This research was supported by the MOTIE (The Ministry of Trade, Industry and Energy), Korea, under the Human Resources Development Program for Convergence Robot Specialists support program supervised by the NIPA(National IT Industry Promotion Agency) (H1502-13-1001).

REFERENCES

[1] http://www.microsoft.com/en-us/kinectforwindows/ (accessed: Jun.12, 2013). Internet, 2013.

[2] Dorit Borrmann, et al., "The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a new Accumulator Design," 3D Research Volume 2 Issue 2, March 2011.

[3] M. A. Fischler and R. C. Bolles, "Random Samples Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," Communicatio-ns of the ACM, Vol. 24, Num. 6, pp. 381-395, June 1981.