Proceedings of the 6th International Conference on Automation, Robotics and Applications, Feb 17-19, 2015, Queenstown, New Zealand

Dominant Plane Detection using a RGB-D Camera for Autonomous Navigation

Jiefei Wang
School of Engineering and
Information Technology
UNSW Canberra
Canberra, Australia 2600
Email: Jiefei.Wang@student.adfa.edu.au

Matthew Garratt
School of Engineering and
Information Technology
UNSW Canberra
Canberra, Australia 2600
Email: M.Garratt@adfa.edu.au

Sreenatha Anavatti
School of Engineering and
Information Technology
UNSW Canberra
Canberra, Australia 2600
Email: S.Anavatti@adfa.edu.au

Abstract—**Dominant plane estimation is an fundamental task not only for trajectory finding problems but also autonomous navigation of mobile robots and MAVs (Micro Air Vehicles). In this paper, we illustrate a novel dominant plane detection approach from a RGB-D camera image sequences. A plane fitting as region growing technique is used in this work, rather than implementing the original algorithm, we modified it and updated to a incremental version, and optimised the plane calculation and mean square error calculation, to improve the accuracy and efficiency. The preliminary experimental results in different scenarios are presented by implementing the algorithm.**

I. Introduction

For the autonomous navigation system of ground mobile robots or MAVs (Micro Air Vehicle), dominant plane detection is a fundamental task for obstable avoidance and trajectory finding problems. The dominant plane can be considered as a planar area where occupies the largest region on the ground towards where the robot moves. It provides useful information about the environment, in particular: whether objects are above the detected dominant plane and along the directon of robot movement and hence can be viewed as obstacles. In this paper, we present a dominant plane detection approach from image sequences captured by a RGB-D camera.

There are a wide range of methods based on vision sensors and ranger sensors which can be used to solve the plane detection problem. In [3, 10], simple approaches to detect the ground floor from a color image were implemented, but can only be applied to very specific environments, such as where a reasonable fraction of colors are needed for detection. In [14], a method was presented which just uses three corresponding points from a stereo camera system to compute the normal vector for a plane. Stereo vision is also used in [6] to recover the structure of the scene. Optical flow field is computed from successive images in [8, 13] to detect the ground plane. This kind of vision based algorithm is not robust enough and computationally complex.

Ranger sensors, such as the laser range finder, provide more simple approaches for plane detection. Some researchers implemented Random Sample Consensus algorithm [11] to detect planar regions by using the images from range sensor [16, 15]. Due to the employment of RANSAC, these algorithms are time consuming. In [2], they employed randomized Hough transform for the real-time plane detection. An alternative approach based on 3D Hough transform to detect the plane from the 3D points clouds is implemented in [12]. The disadvantages of the laser scanner sensor are their high price, and it is seldom feasible to mount them on MAVs due to their limited payload.

Low-cost range sensors are attractive alternatives for expensive laser scanners in many application areas. A recent development in consumer-grade range sensing technology is Microsoft's Kinect sensor (Microsoft. 2010). The Kinect sensor consists of an infrared laser emitter, an infrared camera and an RGB camera. We have selected the Kinect sensor for this work owing to its relatively accurate depth value, its low-cost and the existence of open-source software developed in the robotics community. The open source software is also compatible with the Ubuntu Linux used on our Pioneer robots and Astec Pelican quadrotors making development faster.

In this paper, we illustrate a dominant plane detection approach from a RGB-D camera image sequences. By using a incremental version of the region growing method, the process can be more effective. With some refinement steps applied, the results can be more robust and less time-consuming. The preliminary experimental results are presented by implementing the algorithm in differenct scenarios.

The rest of the paper is organized as follows: Section II describes the Kinect sensor calibration of the IR camera, then convert the depth data of each point into world coordinate system by using the mathematical model. Section III contains the plane model explanation, region growing approach to plane fitting and incremental version for optimal plane calculation. Experimental results for different scenarios are provided and discussed in section IV. The work is summarized in section V.

II. Kinect Sensor Calibration

The Kinect sensor consists of an infrared laser emitter, an infrared camera and a RGB camera. It captures depth and color images simultaneously at frame rates up to 30 Hz. The RGB color camera delivers images at 640 × 480 pixels and 24-bits at the highest frame rate [9]. The 640 × 480 and 11-bits per pixel IR camera provides 2048 levels of sensitivity. The field of view is 50 degrees horizontal and 45 degrees vertical. The operational range of the Kinect sensor is from 50 cm to 400 cm. The IR camera uses a structured light method to measure the depth. An IR emitter projects a known pattern of dots and

978-1-4799-6466-6/15/$31.00 ©2015 IEEE

these dots are recorded by the IR camera and compared to the known pattern.

Fig. 1. Kinect sensor

In order to obtain the actual distance from the Kinect sensor (depth camera) sensor, a further processing is needed. The depth values are measured for a planar surface at several different distances to the sensor using a measuring tape. The actual distance are then plotted against the corresponding normalized disparities observed by the sensor. We extracted Eq. 1 for converting the depth value into actual distance by using the MATLAB curve fitting toolbox. Z is the true depth from the image sensor.

$$Z = a * e^{b*D} + c * e^{d*D} \tag{1}$$

Where $a = 848.7, b = 0.000444, c = -1585, d = -0.0247,$

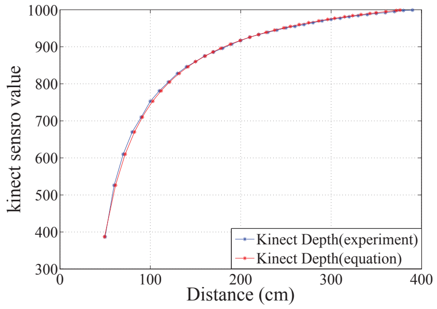


Fig. 2. Kinect depth camera calibration results and line of best fit

and D is the raw depth value from the Kinect sensor. The depth data can be translated into the world coordinate system. The translation can be carried out by using the following equations [7]:

$$X_k = -\frac{Z_k}{f}\left(x_k * \rho_x - x_0 + \delta_x\right) \tag{2}$$

$$Y_k = -\frac{Z_k}{f}\left(y_k * \rho_y - y_0 + \delta_y\right) \tag{3}$$

Where f is the focal lens, the pixel size are ρ_x and ρ_x , the image coordinates are x_k and y_k, the lens distortion are δ_x and δ_y, the principal point offsets are x_0 and y_0. Table 1 shows the parameters of the depth camera which can be used for the calculation of X_k and Y_k.

III. Identifying Planes

A ground plane detection method is described in this section. The plane fitting method is introduced first, then the region growing method is used for detecting the interest points in the image squences from a Kinect sensor. Instead of using the original way for calculation, we optimised the calculation of mean square error to improve the efficiency. The following sections decribe the methodology:

TABLE I. Calibration parameters of infrared camera

Calibration parameter	value
Focal length (f)	4.98 mm
Pixel size (ρ_x)	7.80 μm
Pixel size (ρ_x)	7.80 μm
lens distortion (δ_x)	5.994e-3
lens distortion (δ_y)	-7.027e-4
Principal point offset (x_0)	2.76 mm
Principal point offset (y_0)	2.26 mm

A. Model

Several methods can be used for parametrizing planes. In our work, we chose the Hesse normal form to define a plane. A point r_i from the 3D points data satisfies the following equation is considered located at the plane:

$$n * r_i - d = 0 \tag{4}$$

Where n is the normal vector and d is the distance between the plane and the origin. The normal of the plane is given by the eigenvector corresponding to the smallest eigenvalue of the matrix A:

$$A = \sum_{i=1}^{k} (r_i - r_g)^T \cdot (r_i - r_g) \tag{5}$$

Where

$$r_g = \frac{1}{k}\sum_{i=1}^{k} r_i \tag{6}$$

is the center of the points r_i. Then the optimal plane can be described as :

$$d = n * r_g \tag{7}$$

Fig. 3 shows a plane with three points r_1, r_2, r_3 on it and its normal vector n. Also the distance d between the plane and origin point is presented.

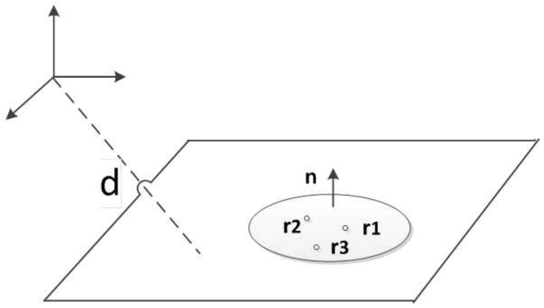


Fig. 3. Plane model

B. Plane Fitting Algorithm

In this section, we extend the work described in [1, 4, 5] by modifying the algorithm and reformulating the underlying mathematics to an incremental version, which allows a highly efficient implementation. This algorithm starts with a randomly chosen point from the 3D points set and applies a region growing technique to find a maximum set of points in the neighbourhood to which a fitting plane can be found. The algorithm proceeds as follows. We first downsample the depth image, since the neighbouring points always carry similar information. Then choose a random point r_1 and point r_2 from a subregion, and treat them as region Π. Then we try to extend the region Π by considering points in increasing distance from Π. Now suppose point r_i is such that the distance between r_i and the region is less than the distance δ. If the mean square error to the optimal plane Ω of the region $\Pi \bigcup r_i$ is less than ϵ and if the distance between the new point and the optimal plane Ω is less than γ, then the r_i is added to the current region Π. We grow this region until no points can be added.

Algorithm 1: Dominant plane extraction algorithm

```
Input: 3D points data;
Select the tuple $r_1$, $r_2$;
$\Pi \leftarrow \{r_1, r_2\}$;
while (new point can be found) do
    Select the nearest neighbours $r_i$ with distance
    $d(\Pi, i) < d$;
    If(MSE($\Pi \bigcup r_i$) $< \epsilon \&\& d(\Pi, i) < d$);
    Then $\Pi \leftarrow (\Pi \bigcup r_i)$;
end
Output: Dominant plane;
```

C. Efficiency improvement

As we mentioned before, it is time consuming if we implement algorithm 1 without any optimization, since there are more than 300,000 points available from the Kinect sensor. The following section describes efficiency improvements that have been made.

- Here we express an incremental version for calculating the plane mean square error (MSE). The MSE is written as below:

  $$MSE = \frac{1}{k}\sum_{i-1}^{k}(n \cdot r_i - d)^2 \tag{8}$$

  where n is the normal vector of the plane, d is the distance between the plane and the origin. Eq.8 can be rewritten as:

  $$MSE = \frac{1}{k} n^T n \sum_{i-1}^{k}(r_i - r_g)^2 \tag{9}$$

  In Eq.9. $\sum_{i-1}^{k}(r_i - r_g)^2$ is matrix A which is the product of the positon covarianc and the number of points. Suppose that s_k is the sum of k points and r_{g_k} is the center of k points. r_{k+1} is a new coming point. We can express an update equation by calculating the difference between A_{k+1} and A_k.

  $$A_{k+1} = A_k + r_{g_k} s_k^T + r_{k+1} r_{k+1}^T - r_{g_{k+1}} s_{k+1}^T \tag{10}$$

  As we can see from Eq.10, it can be easily updated. When a new point r_i add in the region everytime, we only need to calculate the sum of the points rather than the whole process everytime.

- The RGB-D camera alway generate voluminous data that hardly be processed in real time. Due to the fact that close points awlays have similar structure information, we can reduce the data point size rather than use the whole image of 307,200 pixels which is time consuming. In our work, we choose every nth point from every nth row. The sub-sampling factor is 2 which 76,800 points are used.

- From the view of application, mobile robot or MAV starts at a obstacle free area in most situations. If the first point r_1 can be randomly choosen from the bottom half of the image, then it is likely to be on the ground plane. Point r_2 and r_3 are selected from a smaller region based on r_1 randomly. This refinement step implicitly reduces the computation time and improves the efficiency.

- There are some methods to compute neighbours for a point in range image. In our work, we use a k-d tree method to search for nearest neighbors to compute whether they satisfy certain conditions set before. The reason why we chose k-d tree is because it is robust to noise and extract planes fast.

IV. Experimental Results

In this section, by testing our method in several experiments, we discuss the performance for different scenarios of ground plane detection. Both of these experiments use depth image sequences at 30 Hz from Kinect sensor. The results are presented below.

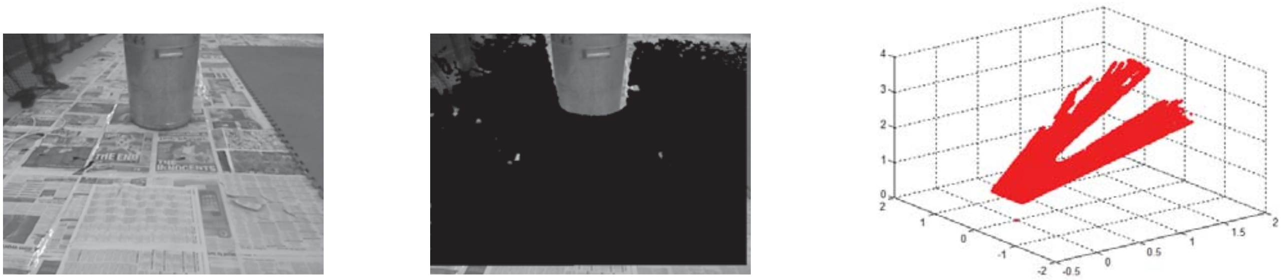
A. Dominant plane detection for a ground platform

In order to do this experiment, a Kinect sensor are mounted on top of the Pioneer-AT robot (see Fig. 6), and desktop hardware with the Ubuntu 12.04 operating system was used. For the first experiment, the Pioneer-AT robot was moved forwards with an obstacle in front of it. The dominant plane (ground plane) is detected and colored by black, and the obstacle (a rubbish bin) is segmented from the ground plane. The result is showed in Fig.4.

B. Dominant plane detection for a flying platform

In order to complete these experiments, the Kinect sensor is held by hand, and moved around the laboratory. There are no obstacles in front of it and the dominant plane is the ground plane shown in Fig.5 (a).

Images of different scenarios are shown in Fig.4 and Fig.5. At the beginning of each experiment, the moving platform is placed at the area where no obstacles in front of it directly, so the seed points can be chosen efficiently. The experimental results showed the dominant planes are detected well compared with their grey image respectively. The operating range of the Kinect sensor is around 50 cm to 400 cm, so if the plane is too close or too far, the results are not reliable. Properites of

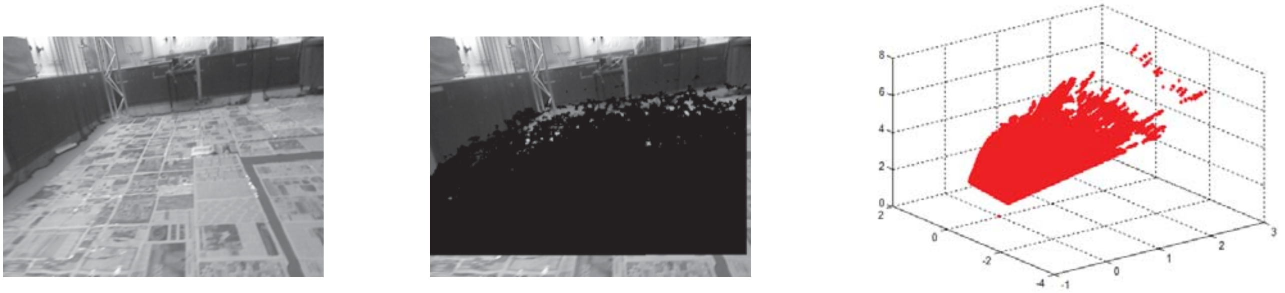


(a) Grey image captured from Kinect sensor

(b) Image with segmented ground plane colored by black

(c) 3D points cloud

Fig. 4. Dominant plane detection graphes from a ground vehicle



(a) Grey image captured from Kinect sensor

(b) Image with segmented ground plane colored by black

(c) 3D points cloud

Fig. 5. Dominant plane detection graphes from a flying platform

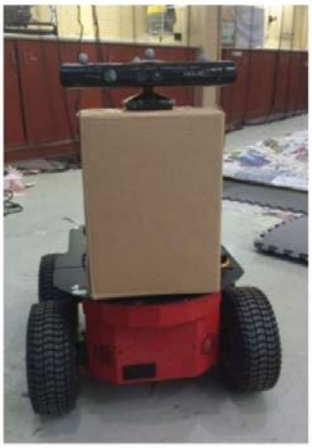


Fig. 6. Kinect sensor mounted on a Pioneer-AT Wheeled Robot

object surfaces also impact on the plane detection. If it is a smooth and shiny surface or occluded, shadowed, then some of the points in that area will become gaps.

V. Conclusion

In this paper, a dominant plane detection alrogithm is proposed by using a RGB-D camera. It can be applied for detecting and avoiding obtacles for a moving platform. In our system, we use a Kinect sensor since its low-cost, low-power, and high resolution. Instead of implementing the original algorithm, we adopt some efficiency improvements and refinement steps to improve the accuracy, and robustness with less computation time.

Experimental results are demonstrated in two scenarios: one is from the view of a ground robot with an obstacle in front of it, another one is moving the Kinect sensor around by hand as if it were mounted on a flying MAV. Detected ground planes are colored by black and 3D point clouds are plotted as well. Futher work will be aimed at implementing this algorithm in real-time obstacles avoidance and simple path planning for a MAV platform such as the Astec Pelican quadrotors.

References

[1] W. Burgard D. Hahnel and S. Thrun. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics band Autonomous Systems*, 44(1):15–27, 2003.

[2] D. Dube and A. Zell. A real-time plane extraction from depth images with the randomized hough transform. In

Workshop on Challenges and Opportunities in Robot Perception, pages 1084–1091, Barcelona,Spain, November 2011.

[3] M. Jungel J. Hoffmann and M. Lotzsch. A vision based system for goal-directed obstacle avoidance. Lisboa, Portugal, 4-5 July 2004. ROBOCUP.

[4] A. Birk J. Poppinga, N. Vaskevicius and K. Pathak. Fast plane detection and polygonalization in noisy 3D range images. In *International Conference on Intelligence and Robots System*, pages 3378–3383, Nice,France, 2008.

[5] Zhang. H Zhang J. Xiao. J, Zhang. J and H. Hildre. Fast plane detection for SLAM from noisy range images in both structured and unstructured environments. In *International Conference on Mechatronics and Automation*, pages 1768–1773, August 2011.

[6] J.-S. Gutmann T. Ohashi K.Kawamoto K. Sabe, M. Fukuchi and T. Yoshigahara. Obstacle avoidance and path planning for humanoid robots using stereo vision. In *International Conference on Robotics and Automation*, volume 1, pages 592–597, New Orleans, April 2004.

[7] K. Khoshelham. Accuracy analysis of Kinect depth data. In *ISPRS workshop laser scanning*, 2011.

[8] Y. Kim and H. Kim. Layered ground floor detection for vision-based mobile robot navigation. In *International Conference on Robotics and Automation*, volume 1, pages 13–18, New Orleans, April 2004.

[9] L. Velho L.Cruz, D.Lucio. Kinect and RGB-D images: Challenges and applications. In *Patterns and Images Tutorials*, pages 36–49, 2012.

[10] S. Lenser and M. Veloso. Visual sonar: Fast obstacle avoidance using monocular vision. In *International Conference on Intelligent Robots and Systems*, number 886-891, 2003.

[11] R. C. Bolles M. A. Fischler. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, June 1981.

[12] R.C. Gonzalez A. Abidi M. Baccar, L.A. Gee. Segmentation of range images via data fusion and morphological watersheds. *Pattern Recognition*, 10:1673–1687, 1996.

[13] N. Ohnishi and A. Imiya. Dominant plane detection from optical flow for robot navigation. *Pattern Recognition Letters*, 27(9):1009–1021, July 2006.

[14] J. Piazzi and D. Prattichizzo. Plane detection with stereo images. In *International Conference on Robotics and Automation*, number 922-927, Orlando, Florida, USA, 2006.

[15] T. Kamegawa T. Fujiwara and A. Gofuku. Evaluation of plane detection with ransac according to density of 3D point clouds. *CoRR*, 25, Sep,2014 2013.

[16] T. Kamegawa T. Fujiwara and A. Gofuku. Plane detection to improve 3D scanning speed using ransac algorithm. In *Conference on Industrial Electronics and Applications*, pages 1863–1869, Melbourne, VIC .Australia, 19-21 June 2013.