Geometry-based 3D Object Fitting and Localizing in Grasping Aid for Visually Impaired

Van-Hung Le*†, Thi-Lan Le*, Hai Vu*, Thuy Thi Nguyen‡, Thanh-Hai Tran*, Tran-Chung Dao§and Hong-Quan Nguyen¶
*International Research Institute MICA, HUST - CNRS/UMI-2954 - GRENOBLE INP, Vietnam
†Faculty of Information Technology, College of Statistics, Vietnam, Email: hunglv@gso.gov.vn
‡Faculty of Information Technology, VietNam National University Agriculture, Vietnam
§University of Information and Communication Technology, Thai Nguyen University
¶Viet - Hung Industrial University

***Abstract*—This paper presents a geometry-based method for 3D object fitting and localization in the context of building a grasping aid service for visually impaired people using information from Kinect sensor. Given two constraints of this working application, (1) the interested object is on a table and (2) the geometrical form of the object is known in advance based on the query of the user, the proposed system consists of three steps: table plane detection, object detection, and object fitting and localization. Our work has three contributions. First, we propose to use organized point cloud representation instead of just point cloud in order to speedup the computational time and improve the accuracy of table plane detection. Second, we employ MLESAC (Maximum LikElihood SAmple Consensus) that can give better results for object fitting. Third, we introduce a new method for evaluating object localization task and make a quantitative evaluation of object localization on our captured dataset.**

I. Introduction

There exists a number of technological tools for assisting visually impaired people [1]. They aim to solve two problems: (1) understanding the current environment and the objects in it, and (2) the self-localization problem. The work presented in this paper belongs to the first category. Our objective is to develop a grasping aid service for the visually impaired people. One of major issues of blind people is to grasp objects without pushing them over. To grasp an object, the following information is needed: the size, the location of the object and a simple description of its shape. For this, the working scenario is described as follows: a visually impaired person wearing a Kinect is going into a kitchen to find and grasp an object on a table. Based on the query from the user, the system can roughly have information about the geometrical structure of the object. For example, a coffee cup usually has cylinder form. Therefore, if the users wish to take a coffee cup, the system should detect all possible cylindrical form objects and give the information (i.e. the center, the radius and the height) so that the users can make corresponding actions.

Recently, with the development of new and low-cost depth sensors such as Microsoft Kinect, this kind of service can be benefited from rich information provided by the depth sensors. While most of current works in the literature follow the appearance-based approach, our works focus on the geometry-based because it can provide directly a simplified description of the objects of interest and it can take advantage of proper characteristic of the working application (objects are on the table and the form of object is known). Moreover, the geometry-based approach can be invariant to object appearance. This means that the system can work with the objects having the same geometrical form with different appearances.

In our previous work, we have proposed a framework for 3D object detection and fitting [2]. The work focused on object detection only. Moreover, it is time consuming and does not provide information about object's location.

In this paper, we extend and improve the previous work with the following contributions. Firstly, we propose to use the organized point cloud in order to speedup the computational time and to give a more accurate table plane and object detection. Secondly, we employ MLESAC (Maximum LikElihood SAmple Consensus) that gives better results of object fitting. Finally, we present an 3D object localization evaluation method and make a quantitative evaluation of object localization on our captured dataset.

II. Related Work

Current works in 3D object finding often focus to solve related problems such as 3D object detection / localization and recognition. Existing methods could be divided into two main categories: appearance-based and geometry-based approaches. The appearance-based approaches do not require explicit description about objects. They try to extract visual features from images or depth map that usually represent implicitly physical properties of the objects. In [3], the authors proposed a viewpoint feature histogram describing 3D point cloud data captured from stereocamera. In [4], the authors used Computer Graphic (CG) CAD models from the Internet and render each CG model from hundreds of viewpoints to obtain synthetic depth maps of the object. For each rendering, a feature vector consisting of point density feature, 3D shape feature, 3D normal features and Truncated Signed Distance Function features are extracted and input to a SVM classifier. However, one of the limitation of this method is how to evaluate on real data from sensors. Drost et al. [5] proposed an approach for 3D object detection using both intensity and depth data. Scale and rotation invariant features are used to describe the

978-1-5090-1801-7/16/$31.00 ©2016 IEEE

objects silhouette and surface appearance. The objects position is determined by matching scene and the model features via a Hough-like local voting scheme. Jerzy et al. [6] has proposed a method for object recognition and localization from 3D point cloud data based on locally calculated feature vectors (FVs). A global descriptor in the form of a set of spatially distributed FVs is created for each reference model. In the detection process, correlation of subsets of reference FVs with FVs calculated in the scene is computed. Then, recognition is based on comparison of the analyzed scene with reference object library.

In geometry-based approaches, knowledge about shape structure of objects of interest should be provided as an explicit model (mostly CAD-like model). Other attributes such as color and texture are usually omitted. This approach is suitable for objects with specific shapes. In [7], the authors propose a method for 3D daily-life object localization using superquadric (SQ) models on point cloud data acquired from Kinect sensor. This method has been tested with Cube-Cylinder object on very simple background that simplifies the object detection. Quantitative evaluation has not been conducted. In [8], multi-scale super-quadric fitting was used to estimate 3D geometric shape and recover pose from unorganized point cloud data. A low latency multiscale voxelization strategy is applied to do the fitting. In our work, we consider kitchenware objects for grasping service. Their shapes are already known, therefore the geometry-based approach is more suitable.

III. 3D Object Detection, Fitting and Localization

In this work, we are interested only in objects on the table. Fig. 1 shows the flow chart of our proposed framework. We first perform table plane detection. Then, we extract only the points belonging to the objects on the table in object detection step. Finally, we process object fitting and object localization in order to determine object shape, size and location. In the following section, we present in detail these steps. It is important to note that our proposed method can work with objects having different geometrical structure. However, in this paper, we focus on only objects with cylinder form.

A. Table plane detection

The table plane detection consists of four steps: down sampling, organized point cloud representation, plane segmentation and plane classification (see Fig. 2). To achieve low computation costs, we reduce data samples in the first step, targeting at a lower sampling rate. However, the sampling rate can not be arbitrarily low because it can significantly affect the subsequent steps and lower the overall detection accuracy. Then, the image data is converted into the organized point cloud data. Each point of the point cloud data has thus a 3-D coordinates (x, y, z) and color values (r, g, b). Using camera intrinsics provided in Microsoft Kinect SDK, each pixel $p(i, j)$ in the RGB image has a color value $C(r_p, g_p, b_p)$ and a depth value $D(x_p, y_p)$ in the corresponding depth image, which can

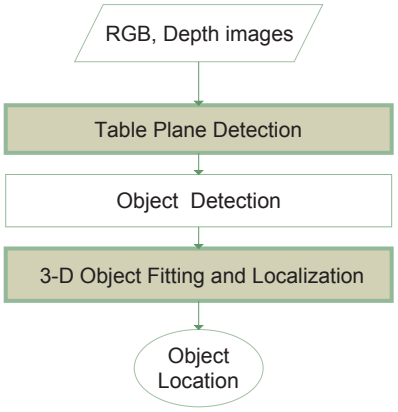

Fig. 1. Proposed approach for 3D object detection, fitting and localizing in object grabbing aid service for visually impaired people.

be projected into the metric 3-D space using the following Eq. 1:

$$x = \frac{z(x_p - c_x)}{f_x}; \quad y = \frac{z(y_p - c_y)}{f_y}; \quad z = D(x_p, y_p); \tag{1}$$

with (f_x, f_y), (c_x, c_y) being the focal length and principal point respectively.

The organized point cloud data follows the structure of a matrix as in the image. Each point has a 2-D index (i, j). Here, (i, j) are the indices of the row and column of the matrix respectively. They are limited by the size of the obtained image by the sensors. For example, the image obtained from the Microsoft Kinect camera has 640x480 pixels, then $i = 1, ..., row$; $j = 1, ..., col$; normally with (row, col) = $(480, 640)$. Matrix P presents the organized point cloud data of a scene based on Eq. 2:

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,3} & p_{1,4} & \cdots & p_{1,col} \\ p_{2,1} & p_{2,2} & p_{2,3} & p_{2,4} & \cdots & p_{2,col} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{row,1} & p_{row,2} & p_{row,3} & p_{row,4} & \cdots & p_{row,col} \end{bmatrix} \tag{2}$$

where $p_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$, $(x_{i,j}, y_{i,j}, z_{i,j})$ are the values of point $p_{i,j}$ in 3-D space.

For plane segmentation, RANSAC or one of its variants can be used. However, this step requires highly accurate and real-time plane extraction. Therefore, we employ the plane segmentation method of [9] that combines distance and normal vector information.

To select the table plane among the extracted planes, in the plane classification step, we adopt the accelerometer data from the Kinect camera. The main constraint is that the table should stand on the floor. Therefore a table plane should be parallel with the floor plane. The accelerometer data provides us the normal vector of the ground (floor) plane and other planes, that are parallel with the table planes. We eliminate planes which do not meet this criteria. The remaining planes are considered

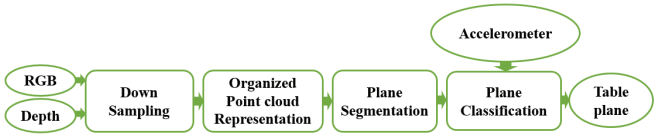

Fig. 2. Table plane detection flow chart.

as table plane if they are high enough. After the table plane was detected, we separate the data of the objects on the table based on the table plane's information.

B. 3D object fitting and localization

1) 3D object fitting: The outputs of the object detection are a set of clusters of point cloud data. Each cluster corresponds to an object. For each cluster, we can apply a fitting method in order to estimate the object information. Currently, there are a number of algorithms that can be used for estimating a model from a set of points such as RANSAC [10], RANSAC variants [11], Least Squares [12]. In [11], MLESAC algorithm has been proved to be robust. Therefore, in this paper, we use MLESAC for object fitting. In the RANSAC algorithm, if the threshold T for determining inlier points is set too high then the robust estimation result can be very poor. It finds the minimum of a cost function C as Eq. 3.

$$C = \sum p(e_i^2) \tag{3}$$

where $p(e_i^2)$ is the error of a sample i^{th} with the estimated model. $p(e^2)$ is determined as Eq. 4.

$$p(e^2) = \begin{cases} 0 & e^2 < T^2 \\ constant & e^2 \geq T^2 \end{cases} \tag{4}$$

For fast convergence of error function, Torr et al. [13] proposed the maximum likelihood sample consensus (MLESAC). Based on the work of Torr et al., error is modeled as a mixture model of Gaussian and is computed as Eq. 5.

$$C = \sum p(e_i^2) \tag{5}$$

Therein, model error as mixture model is defined as

$$p(e) = \left(\gamma\left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n exp\left(-\frac{e^2}{2\sigma^2}\right) + \left(1-\gamma\right)\frac{1}{v}\right) \tag{6}$$

where

$$p_{inlier} = \left(\gamma\left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n exp\left(-\frac{e^2}{2\sigma^2}\right)\right)$$
$$p_{outlier} = \left(\left(1-\gamma\right)\frac{1}{v}\right) \tag{7}$$

where σ is the standard deviation of the error; v is parameter space within which outliers are expected to fall; γ is the mixing parameter that it is estimated based on Expectation Maximization (EM) from a set of indicator variables η_i with $(i = 1, ..., n)$. It is presented the detail in [13].

The minimized error $-L$ is the negative log likelihood:

$$-L = -log\left(\gamma\left(\frac{1}{\sqrt{2\pi\sigma}}\right)^n exp\left(-\frac{e^2}{2\sigma^2}\right) + \left(1-\gamma\right)\frac{1}{v}\right) \tag{8}$$

The function $-L$ is used as a score for determining the error function in MLESAC algorithm.

$$p(e_i^2) = \begin{cases} -L & e^2 < T^2 \\ T^2 & e^2 \geq T^2 \end{cases} \tag{9}$$

The MLESAC algorithm for estimating a model is summarized as follows:

- randomly select a smallest possible subset of data p_i for creating a model m_i
- test the data against model m_i, expand hypothetical inliners with all points satisfy a threshold T
- reestimate the model m_{i+1} with all points supporting the model
- calculate the cost function -L (see Eq. 8) of the newest model
- repeat and keep the model with lowest error

We employ MLESAC as a fitting method in two cases as follows:

Case 1: MLESAC is applied on 3D point cloud data. In this case, we estimate a cylinder.

Case 2: MLESAC is applied on the projected data on the table plane. In this case, we estimate a circle corresponding to the bottom part of the object.

The output of this step is object's location (the coordinates of object center) and the radius of the estimated cylinder or circle.

2) 3D object localization: In the previous step, we can get object position in Kinect coordinate system. However, in order to give object location to the users and to evaluate object localization methods, we have to convert this coordinate to a predefined coordinate system. In order to do this, we design a pattern similar to a chess board on the table. In our experiment, the size of each cell of the pattern is 10cm because with that size one can clearly see the chessboard on the table plane from 1.5m (the distance between Kinect and the table plane). On the chessboard, we define a coordinate system as illustrated in Fig. 3a. To transform from Kinect coordinate system to chessboard coordinate system, we have to find the transformation matrix (rotation and translation matrix). In [14], Horn et al. stated that the minimum number of 3-D points necessary for estimating the rotation and translation parameters is 4. The more number of points is selected, the lower error is. In this paper, we select 12 points. The error of transformation estimation is defined as

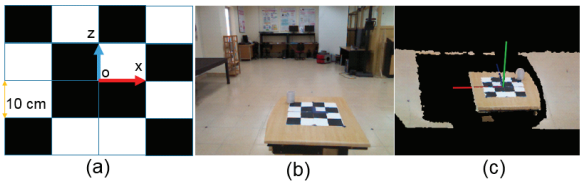

Fig. 3. (a) Predefined coordinate system (similar to a chess board) on the table; (b) RGB image; (c) Point cloud of the scene.

Eq. 10.

$$Err = \frac{1}{12}\sum_{i=1}^{12}\sqrt{(x_s - x_t)^2 + (z_s - z_t)^2} \qquad (10)$$

where (x_s, z_s) is the source point; (x_t, z_t) is the target point. After determining the transformation matrix, we convert all point clouds from Kinect coordinate system to chessboard coordinate system (see Fig. 3c).

IV. Experiment

We implement the proposed approach using C++ language and two libraries ()PCL 1.7 and OpenCV 2.4.9) on a PC with Core $i5$ processor and 8G RAM.

A. Table plane detection evaluation

In order to evaluate table plane detection, we capture images of 10 scenes. Some examples of the captured scenes are shown on Fig. 4. The total frames used for table plane detection evaluation is 6686 frames. The results show that the table

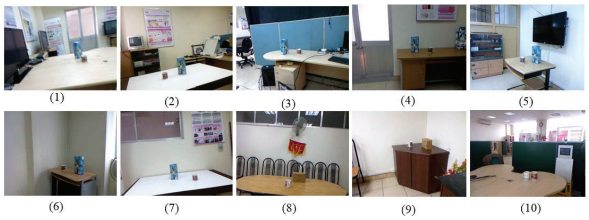

Fig. 4. Examples of 10 scenes captured in our dataset.

TABLE I
THE AVERAGE RESULT OF DETECTED TABLE PLANE OF OUR METHOD WITH DIFFERENT DOWN SAMPLING FACTORS.

Down sampling	Average recall (%)	Frame rate
Without downsampling	97.52	0.83
(3x3)	97.00	5
(5x5)	92.21	14
(7x7)	84.13	33

plane detection based on organized point clouds obtain not only good result in term of recall but also computational efficiency. It is important to note that our previous works based on geometry constraints [2] gets 82.15% of recall with 0.2 frames per second on this dataset. Using the down sampling with size (3x3) (the depth value of a center pixel in the depth image is the average depth value of the three pixels neighboring), we can obtain results as good as without down sampling (the average recalls are 97% and 97.52% respectively). Our system can process 5 frames per second while the later case can only process 0.83 frames per second. If we increase the down sampling size, the frame rate will increase while the recall will decrease. Therefore, for a trade off, in this paper we use a size of (3x3) for down sampling for table detection. Fig. 5 shows some examples of table plane detection results. In our data set, there is only one table plane in each scene.

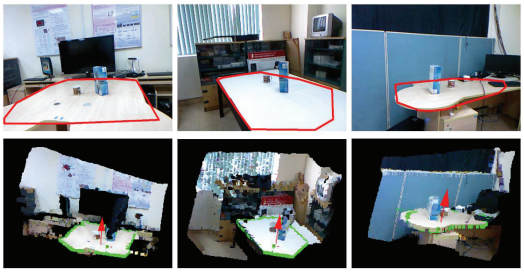

Fig. 5. Top line is some results of table plane detection step. The detected table planes are marked by the red color boundary; Bottom line is some results of table plane detection in the point clouds, table plane is marked by green color points (3-D contour), red color vector is the normal vector of the table plane.

B. 3D object fitting and localization evaluation

1) Object localization measure: We evaluate the performance of object localization method by two measures. The first measure is the error ε between the estimated center $C_e(x_e, y_e, z_e)$ and the predefined object's center location $C_g(x_g, y_g, z_g)$. This is determined as Eq. 11.

$$\varepsilon = \sqrt{(x_g - x_e)^2 + (y_g - y_e)^2 + (z_g - z_e)^2} - Err \qquad (11)$$

where Err is defined in Eq. 10.
The second measure is the difference between the estimated radius and the ground-truth radius.
2) Results: In our experiments, we use four cylinder cups without handle. The cups are different in appearance, radius and height. Images of these cups are shown in Fig. 6. We perform 3 experiments that are presented in detail in the following.

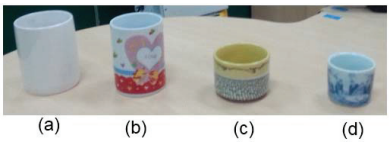

Fig. 6. Images of four cups (from left to right: Object 1, Object 2, Object 3, Object 4) with cylinder form using in our experiments.

Experiment 1 (Exp1): In this experiment, Kinect is mounted on a shelf. The distance between Kinect and table center is about 1.5m. The height between Kinect and the table plane is about 0.6m. Kinect is set at four positions around the table (see Fig. 7a). At each capture time, we put only one object on the table. The locations of the object on the chessboard are (-2, 0, 2), (2, 0, 0), (0, 0, -2), (-2, 0, 2). Therein, each cell is corresponding to a unit (10cm).
Experiment 2 (Exp2): In this experiments, the Kinect is mounted a person's chest and the person is moving around the table (see Fig. 7b). The locations of all objects on the chessboard is (-2, 0, 0).
Experiment 3 (Exp3): This experiment is similar to the first one. However, for each capture time, we put more than one

Fig. 7. Setup of (a) the experiment 1 and (b) the experiment 2.

object (two or three) at different locations. The minimum distance between two objects is greater than 10cm.
The number of captured frames in experiment 1, 2 and 3 is 452, 600 and 125 respectively.

The average (ε) and standard deviation (θ) of error and its distribution obtained in the experiment 1 and the experiment 2 for two cases (Case 1: cylinder estimation and Case 2: circle estimation) are shown as Tab. II and Fig. 8 respectively. We can see that the average error of both cases is small (2.1423 and 1.2683 cm). In both experiments, the second case always gets better results in term of average value. However, it is less stable than the first case. In term of radius estimation, we compare the estimated radius in two cases with ground-truth information (see Tab. III). The difference between the ground-truth and the estimated radius in both cases is less than 0.5cm. Once again, the second case shows that it can estimate better the radius of the objects.

TABLE II
THE AVERAGE (ε) AND STANDARD DEVIATION (θ) OF ERROR OF THE OBJECT CENTER ESTIMATION IN EXPERIMENT 1 AND EXPERIMENT 2.

Case	Average (ε)(cm)		Standard deviation (θ)	
	Exp 1	Exp 2	Exp 1	Exp 2
Case 1	2.1423	1.7283	0.1545	0.9746
Case 2	1.2683	1.2834	0.9727	1.2677

TABLE III
THE AVERAGE RADIUS OF THE ESTIMATED OBJECT (CM) FROM EXPERIMENT 1 AND EXPERIMENT 2

Case	Average radius (cm)			
	Object 1	Object 2	Object 3	Object 4
Experiment 1				
Case 1	3.53	3.07	3.29	2.47
Case 2	3.67	3.22	3.42	2.95
Ground-truth	3.75	3.50	3.75	3.00
Experiment 2				
Case 1	3.42	3.21	3.33	2.47
Case 2	3.35	3.45	3.6	2.69
Ground-truth	3.75	3.50	3.75	3.00

In the first two experiments, we put only one object on the table. The main purpose of the third experiment is to evaluate the proposed method in the scenario with more than one object on the table. We capture 125 frames of 3 different scenes containing two or three objects. The result of object detection

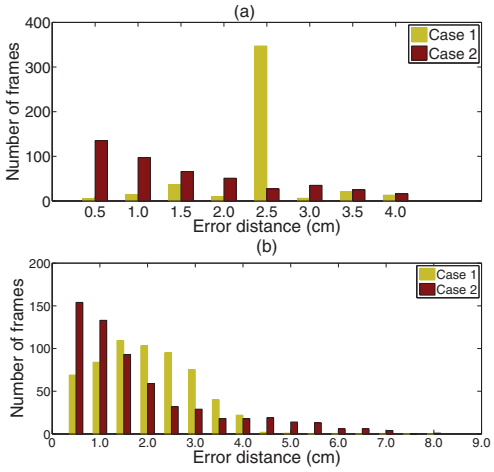

Fig. 8. The distribution of error (in distance measure) (ε) of object center estimation in two cases (Case 1: cylinder and Case 2: circle estimation) obtained from experiment 1 (a), and the experiment 2 (b).

step is shown on Tab. IV. We can observe that, the proposed method can detect objects with nearly 90% of recall. However, in some cases, since some parts of the table plane are missed therefore the objects are not well detected.

TABLE IV
THE RESULT OF OBJECT DETECTION STEP IN EXPERIMENT 3.

Scene	Scene 1	Scene 2	Scene 3
#Frame	72	32	21
Number of object instances	144	64	63
Number of detected object instances	134	56	60
Recall (%)	93.05	87.5	95.23

Concerning object fitting and localization step, we evaluate object fitting and localization for each object on the table. The average error distance (ε) and object's radius of experiment 3 are shown on Tab. V, while the distribution of the error for each object is illustrated on the Fig. 9. The average value of the error distance (ε) for three objects is smaller than 2.5 cm. Concerning the computational time, the average time of

TABLE V
THE AVERAGE ERROR (σ) AND THE RADIUS OBTAINED FOR THREE OBJECTS OF EXPERIMENT 3.

Object/Case	Object 1	Object 2	Object 3
Average value of error distance (ε) (cm)			
Case 1	1.8483	2.4427	2.5224
Case 2	1.7583	2.3379	2.3958
Average radius (cm)			
Case 1	3.37	3.17	3.24
Case 2	3.48	3.46	3.52
Ground-truth	3.75	3.50	3.75

object fitting and localizing step are 0.06s/object for case 1 and 0.02s/object for case 2. The average processing time of all steps are 0.36s/frame for case 1 and 0.42s/frame for case 2. Therefore, the case 2 can get better result in term of the

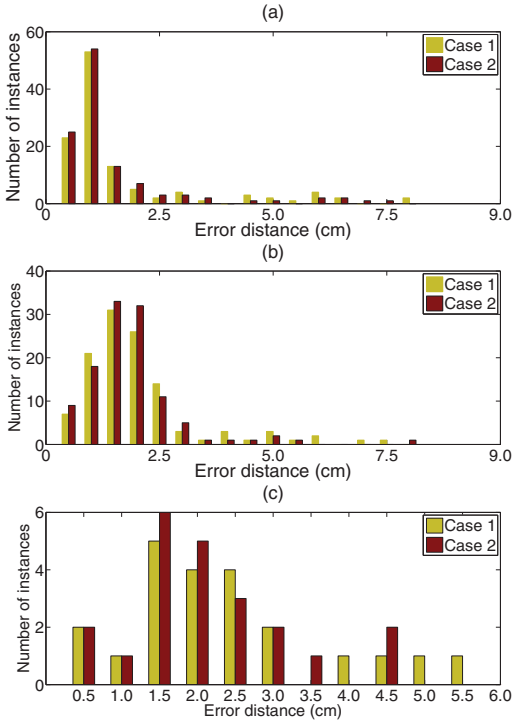

Fig. 9. The distribution of error distance (ε) obtained for 3 objects in the experiment 3; (a) Object 1; (b) Object 2; (c) Object 3.

average of errors and is faster than case 1. However the first case is more stable.

V. Conclusion

In this paper, we have proposed a geometry-based method for 3D object fitting and localization in the context of object grasping aid service for visually impaired people. The experimental results on table plane detection show that the use of down sampling and organized point cloud allows to perform table plane detection with 97% of recall and 5 frames per second. This improves largely the performance of table plane detection and then can improve the performance of the overall system. Concerning object fitting and localization, the proposed method can estimate the object's center position with the an error less than 2.5 cm and the radius with an error of less than 0.5cm. In this paper, we just evaluated the proposed method with objects having the cylinder form. Moreover, the number of objects on a table is limited and the computational time is still high. In the future, we will work with different objects forms and try to reduce the computational time.

Acknowledgements

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number FWO.102.2013.08.

References

[1] Marion Hersh and Michael A. Johnson. *Assistive Technology for Visually Impaired and Blind People*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[2] Van-Hung Le, Hai Vu, Thuy Thi Nguyen, Thi-Lan Le, Thi-Thanh-Hai Tran, Michiel Vlaminck, Philips Wilfried, and Peter Veelaert. 3D Object Finding Using Geometrical Constraints on Depth Images. The 7th International Conference on Knowledge and System Engineering, 2015.

[3] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux, and John Hsu. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram. pages 2155 – 2162, 2010.

[4] Shuran Song and Jianxiong Xiao. Sliding Shapes for 3D Object Detection in Depth Images. ECCV2014, 2014.

[5] Bertram Drost and Slobodan Ilic. 3D Object Detection and Localization using Multimodal Point Pair Features. 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012.

[6] Jerzy Bielicki and Robert Sitnik. A method of 3D object recognition and localization in a cloud of points. *EURASIP Journal on Advances in Signal Processing*, pages 1–13, 2013.

[7] Ilya Afanasyev, Nicolo Biasi, Luca Baglivo, and Mariolino De Cecco. 3D Object Localization using Superquadric Models with a Kinect Sensor. *PCL Toyota Code Sprint Final Report - Smoothing*, 2012.

[8] Kester Duncan, Sudeep Sarkar, Redwan Alqasemi, and Rajiv Dubey. Multi-scale Superquadric Fitting for Efficient Shape and Pose Recovery of Unknown Objects. In *ICRA*, 2013.

[9] S.H. Dirk Holz, Radu Bogdan Rusu, and Sven Behnke. Real-Time Plane Segmentation Using RGB-D Cameras. *RoboCup 2011: Robot Soccer World Cup XV Volume 7416 of the series Lecture Notes in Computer Science pp 306-317*, 2011.

[10] M. A. Bolles and R.C. Fischler. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysisand Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[11] Sunglok Choi, Taemin Kim, and Wonpil Yu. Performance Evaluation of RANSAC Family. pages 81.1–81.12. British Machine Vision Association, 2009.

[12] H. Badino. Least Squares Estimation of a Plane Surface in Disparity Image Space. *Carnegie Mellon University Pittsburgh, PA 15217, USA*, 2011.

[13] Philip H. S. Torr and Andrew Zisserman. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[14] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.