Fitting Cylindrical Objects in 3-D Point Cloud using Contextual and Geometrical Constraints*

HAI VU[1*], VAN-HUNG LE[1,2], THI THUY NGUYEN[3],
THI-LAN LE[1], THANH-HAI TRAN[1]
[1]*International Research Institute MICA, HUST - CNRS/UMI-2954 -GRENOBLE INP, Vietnam*
[2]*Tan Trao University, Vietnam*
[3]*Faculty of Information Technology, Vietnam National University of Agriculture, Vietnam*
E-mail: {hai.vu,van-hung.le,thi-lan.le,thanh-hai.tran }@mica.edu.vn

In this paper, we propose a framework for fitting cylindrical objects toward deploying an object-finding-aided system for visually impaired people. The proposed framework consists of a RANSAC-based algorithm and a model verification scheme. The proposed robust estimator named GCSAC (Geometrical Constraint SAmple Consensus) avoids expensive computation of the RANSAC-based algorithms due to its random drawing of samples. To do this, GCSAC utilizes some geometrical constraints for selecting good samples. These constraints are raised from real scenarios or practical applications. First, the samples must ensure being consistent with the estimated model; second, the selected samples must satisfy explicit geometrical constraints of the interested objects. In addition, the estimated model is verified by using contextual constraints, which could be raised from a certain scene such as object standing on a table plane, size of object, and so on. GCSAC's implementations are carried out for various estimation problems on the synthesized dataset. The comparisons between GCSAC and MLESAC algorithm are implemented on three public datasets in terms of accuracy of the estimated model and the computational time. Details of algorithm implementation and evaluation datasets are publicly available.

Keywords: Primitive Shape Estimation, RANSAC Variations, Quality of Samples, 3D Point Cloud, Cylinder fitting for finding

1. INTRODUCTION

Estimating parameters of a primitive geometric shape such as plane, sphere, cylinder or cone from a 3-D point cloud data is a fundamental research topic in the fields of computer vision and robotics. The geometrical model of an interested object can be estimated using from two to seven geometrical parameters [1]. Random Sample Consensus (RANSAC) algorithm [2] and its paradigm attempt to extract as good as possible shape parameters. These algorithms face critical issues such as heavy noise of the data or processing time constraints. Originally, the RANSAC algorithm consists of hypotheses based on drawing randomly 3-D points from an input data set. Although its variants such as PROSAC algorithm [3] proposed a so-called guided sample schemes, one still needs more investigation to efficiently adapt such techniques to real scenarios. In practice, a *priori* knowledge is often available in many applications. We can exploit this knowledge to generate better hypotheses, then a better model is estimated. In this paper, we propose a new RANSAC-based algorithm, named Geometrical Constraint SAmple Consensus - GCSAC. The proposed algorithm is inspired by guiding the

Received July 8, 2017; revised Jan. 1, 2018; accepted Jan. 15, 2018.
Communicated by: Chu-Song Chen

minimal subset selection using normal constraints of geometric models and the context of scene. We demonstrate GCSAC for fitting cylindrical objects in a real application that aims to support visually impaired people in some daily activities.

Fig. 1. Illustration of the service supporting a visually impaired person to find an object in a cafeteria room.

Let us consider a real scenario that usually happens in daily life of a visually impaired person. The visually impaired looks for a cup to drink water, he/she gets into the kitchen and touches any surrounded object to pick up the right one. Our system could help the visually impaired person as illustrated in Fig. 1. The person just makes a query “*Where is the coffee cup?*” Knowing the object of interest is a cup which usually have cylindrical shape, the system captures the 3D the scene by a RGB-D sensor, then fits a primitive shape (e.g. cylindrical shape) from the collected 3-D point cloud data. The *prior* knowledge observed from the current scene could be the coffee cup should have cylinderical shape and normally stands on the table. Other contextual constraints could be walls in the scene are perpendicular to the table plane; the size of the table plane is limited. These constraints raise many cues for improving the RANSAC-based approaches. In this paper, we exploit geometrical constraints of the interested object to search qualified samples and contextual constraints to verify the estimated model in a post-processing step. By this way, our proposed GCSAC overcomes main issues of original RANSAC in term of quality of the estimated model and the computational time.

In this paper, we first describe the GCSAC for fitting a query-based object like a cylindrical model. Instead of randomly drawing 3-D points from an input data set without any prior assumption on the data as RANSAC paradigms, at each hypothesis of GCSAC, we select *good samples* based on the constraints of the estimated model. To select good samples, we define two criteria: (1) the selected samples must ensure being consistent with the estimated model via a rough evaluation of inlier ratio; (2) the selected samples must satisfy some explicit geometrical constraints of the specific shape (e.g., cylindrical constraints). The idea is that at each iteration, thanks to the good samples, the better model (with higher inlier ratio) is highly expected. Consequently, the number of iterations can be adaptively updated according to the inlier estimation. The estimation procedure achieves an earlier termination.

To evaluate the sample consensus, GCSAC utilizes a Negative Log-Likehood criterion as defined by MLESAC algorithm [4]. Moreover, the estimated model is verified using contextual constraints. In the experimental evaluations, the proposed algorithm is compared with the original MLESAC on three realistic datasets in terms of both quality of the estimated model and the computational time. These datasets consist of cylindrical objects that are collected from various practical scenes.

2. RELATED WORK

For a general introduction and performances of RANSAC family, readers can refer to good surveys in [5, 6]. In the context of this research, we briefly survey related works which are categorized into two topics. First, efficient schemes on the selection of minimal subset of samples for RANSAC-based robust estimators; and second, techniques for estimating parameters of primitive shapes, particularly, focusing on estimations of cylindrical objects.

For the first topic, the original RANSAC with a straightforward implementation always requires considerable computational time. Many RANSAC variants have been proposed with further optimization for a minimal sample set (MSS) selection. Progressive Sample Consensus or PROSAC [3] orders quality of samples through a similarity function of two corresponding points in the context of finding good matching features between a pair of images. In PROSAC algorithm, the most promising hypotheses are attempted earlier; therefore drawing the samples is implemented in a more meaningful order. However, PROSAC faces critical issues for defining the similarity function. LO-RANSAC [7] and its fixed version LO$^+$-RANSAC [8] add local optimization steps within RANSAC to improve accuracy. To speed up the computation, adaptive RANSAC [9] probes the data via the consensus sets in order to adaptively determine the number of selected samples. The algorithm terminates immediately at a smaller number of iterations. With the proposed method, the *good samples* are expected to generate the best model as fast as possible. Therefore, the termination condition of the adaptive RANSAC [9] should be explored. In the field of 3-D object recognition, Chen et al. [10] propose DARCES (Data Aligned Rigidity Constrained Exhaustive Search) that deals with a partial matching problem and combines the rigidity constraint from the pre-selected control points. This constraint has been applied to a partial object to reduce processing time. In contrast, Drost et al. [11] proposed a framework (and Tolga et al. [12] revisited) that creates a global model description based on pair of feature points. Matching a model to a scene utilizes an efficient Hough-like voting scheme on a reduced pose parameter space. Recently, USAC [13] introduced a new framework for a robust estimator. In USAC framework, some strategies such as the sample check (Stage 1b) or the model check (Stage 2b), before and after model estimation, respectively, are similar to our ideas in this work. However, USAC did not really deploy an estimator for primitive shape(s) from a point cloud. A recent work [14] proposed to use geometric verification within a RANSAC framework. The authors deployed several check procedures such as sample relative configuration check based on the epipolar geometry. Rather than the “check” procedures, our strategies anticipate to achieve the best model as soon as possible. Therefore, the number of iterations is significantly reduced thanks to the results of searching for good

sample process.

For cylindrical object estimation (or more general, fitting primitive shapes) from 3-D point clouds, readers can refer to a survey on feature-based techniques [15]. Some fitting techniques, for instance, multiscale super-quadric fitting in [16], Hough transform in [17], are commonly used. However, the robust estimators (e.g., RANSAC family [6]) are always preferred techniques. Original RANSAC [2] demonstrates itself robust performances in estimating cylinders from range data. In [18], normal vectors and curvature information are used for parameters' estimation and extraction of cylinders. The cylindrical objects are also interested in the analytic geometrical techniques. The authors in [19] formulate a cylinder using three parameters such as radius r, height h, and axis γ. [1] defines a cylinder through two samples and their normal vectors. In this study, geometrical analysis of a cylinder in [1] is adopted for defining criteria of the qualified samples as well as for estimating parameters of the interested model from a 3-D point cloud.

3. PROPOSED METHOD

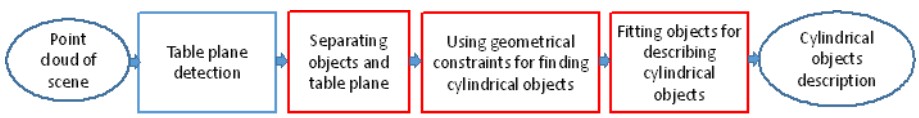


Fig. 2. The proposed framework for fitting cylindrical objects using contextual and geometrical constraints.

3.1. Proposed general framework for fitting cylindrical objects

In the context of developing an aided-system for visually impaired people (as shown in Fig.1), we propose a framework which consists of four main steps as presented in the Fig. 2. The step of table plane detection, presented in our previous work [20], has achieved a high accuracy and real time performance by combining depth and acceleration features from the Microsoft Kinect sensor. In this paper, we describe remaining steps of the framework (marked in red blocks) in the Fig. 2.

3.2. Separating objects from the table plane

Given a normal vector $\mathbf{n}_t$ of the table plane in a current scene, the original point cloud data is transformed into a new coordinate system. This step facilitates the computation and the use of geometric constraints. Let's denote the Kinect's original coordinates $O_k(x_k y_k z_k)$, the new coordinate system is specified by a new origin O_t (x_t, y_t, z_t) and its normal vector. The transformation between two coordinate systems is described by rotation and translation matrices. The rotation matrix is:

$$R_x(\alpha) = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{vmatrix} \quad (1)$$

$$R_y(\beta) = \begin{vmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{vmatrix} \quad (2)$$

$$R_y(\Theta) = \begin{vmatrix} \cos(\Theta) & \sin(\Theta) & 0 \\ -\sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad (3)$$

where α, β , Θ are rotation angles in x, y, z axis.
From Eq. (1), Eq. (2) and Eq. (3), we have the rotation matrix:

$$R = Rz(\Theta)Ry(\beta)Rx(\alpha) \quad (4)$$

In this context, the transformation is defined by a rotation and a translation. The rotations in x-axis, z-axis are defined by angle α,Θ:

$$\alpha = \text{acrsin}\frac{b}{\sqrt{(b^2+c^2)}} \qquad \Theta = \text{acrsin}\frac{c}{\sqrt{(c^2+a^2)}} \quad (5)$$

where (a,b,c) are components of the normal vector $\mathbf{n}_t$ of table plane. We perform a translation *d* in y-axis by a term:

$$d = |yt| \quad (6)$$

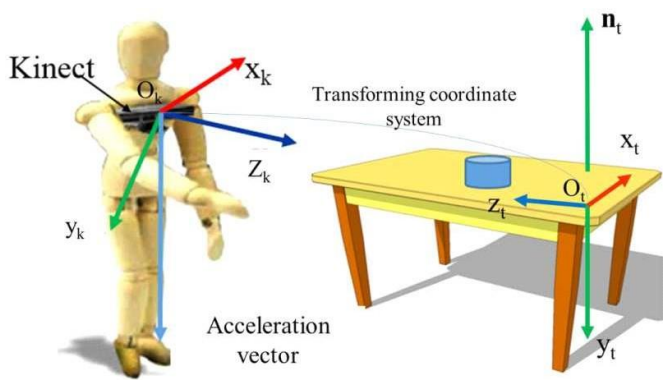


Fig. 3. The transformation of original point cloud: from Kinect's original coordination $O_k(x_k,y_k,z_k)$ to a new coordination $O_t(x_t,y_t,z_t)$, the normal vector $\mathbf{n}_t$ of a table plane is parallel to the y-axis.

The transformation allows to filter data points from original scene. The data points whose y-values are smaller than the minimum y-values of points belong to the table plane as shown in Fig. 3. They will be preserved based on an assumption that the query objects are always on the table. In addition, it allows us to use geometrical constraints such as the height of the object, the difference between main direction of the cylinder and the table plane's normal vector should small enough. In the next step, the point cloud of a

cylindrical object is fitted using the proposed GCSACalgorithm.

3.3. The proposed robust estimator - GCSAC

To estimate parameters of a primitive shape, RANSAC-based algorithms (RANSAC, MLESAC, MSAC, LOSAC) usually draw randomly a Minimal Sample Set (MSS) or semi-random (PROSAC) or using constraints of the sample's distribution (NAPSAC). The proposed GCSAC constructs a MSS in a different manner where random sampling procedures aims at probing the consensus data to be easily achievable. To do this, a low inlier threshold is pre-determined. After only (few) random sampling iterations, the candidates of good samples could be achieved. Once initial MSS is established, its samples will be updated by the qualified one (or good sample) so that the geometrical constraints of the interested object is satisfied. The estimated model is evaluated according to Maximum Log-likelihood criteria as MLESAC [4]. The final step is to determine the termination condition, which is adopted from the adaptive RANSAC algorithm [9]. Once the higher inlier ratio is obtained, the criterion termination K for determining a number of sample selections is updated by:

$$K = \frac{\log(1-p)}{\log(1-w^m)} \tag{7}$$

where p is the probability to find a model describing the data, s is the minimal number of samples to estimate a model, w is percentage of inliers in the point cloud. While p often set to a fixed value (e.g., $p = 0.99$ as a conservative probability), K therefore depends on w and m. The algorithm terminates as soon as the number of iterations of current estimation is less than that has already been performed.

Algorithm 1: GCSAC's implementation for fitting a cylindrical object from the point cloud

Input: 3D Points with normal vectors: U_n,$U\ n_n$; iterations K
Output: Estimated parameters of the cylinder;

Algorithm:
Step 1: initialization
Step 2: While ($k < K$)
{
2.1 $k++$;
Drawing randomly two points $P = \{p_1, p_2\}$ from U_n;
2.2 $U^*_n = \emptyset$;
2.3 if (U^*_n != $\emptyset$) //estimate model M_k from P else goto 2.1
2.4 Compute w_k
2.5 if ($w_k \geq w_t$) and ($w_k > w_m$){
Search p^*_2 by (9);
Update $U^*_n = \{p_1, p^*_2\}$
$w_m = w_k$;}
2.6 Re-estimate M_k from U^*_n
2.7 Compute $A_d = (\gamma_c, \mathbf{n}_t)$
2.8 if ($A_d < A_t$) compute $-L$ else goto 2.1

2.9 if ($-L < L_t$){
choose the best model M_b;
re-compute K;}
else goto 2.1
}

Details of the GCSAC's implementation are given in Algo. 1. Obviously, the criterion defining the good samples is the most important. Based on the idea of adaptive RANSAC [9] to probe initial samples, GCSAC starts from roughly selecting initial good samples. To initialize the stack U^*_n, where U^*_n is used to store $m - 1$ sample points and its inlier ratio w_i), we assume that the worst case of inlier ratio $w_t = 0.1$ (10%) is determined. Therefore, a consensus set containing more than 10% of the data is easily found. A model is estimated from m random samples. In case of estimating a cylinder, m equals 2 ($m = 2$ [1]).

After that, U^*_n is reset. m samples and the inlier ratio w_i of the estimated model are stored in U^*_n if w_i is equal to or greater than w_t. Then, the MSS utilizes $m - 1$ kept *good* samples. The remaining m^{th} sample will be replaced by a better one which best satisfies the *geometrical constraints* of the interested shape. The good samples that satisfy the geometrical principles of a primitive shape are explained in Section 3.4. If none of iterations satisfies $w_i \geq w_t$, the estimation algorithm degrades to the original MLESAC. The inlier ratio of a iteration depends on the threshold T. The selection of an optimal value for T is out of scope of this research.

3.4. Geometrical analyses for qualifying good samples

A cylinder is determined by following parameters: a center point on the cylinder axis, denoted as $I_c(x_0, y_0, z_0)$; the main axis direction is a vector, denoted γ_c; and its radius R_c. For geometrical analyses of a cylinder object, we adopted the analysis given by [1]. Using this setting, a cylinder is estimated from two points (p_1, p_2) (two grey-squared points in Fig. 4(a)) and their corresponding normal vectors ($\mathbf{n}_1$, $\mathbf{n}_2$) (the blue lines in Fig. 4(a)). Let γ_c is the main axis direction (the pink line in Fig. 4(a)) of the cylinder. It is estimated by $\gamma_c = \mathbf{n}_1 \times \mathbf{n}_2$. To estimate a centroid point I_c, we project two parametric lines $L_1 = p_1 + t\mathbf{n}_1$ and $L_2 = p_2 + t\mathbf{n}_2$ along the axis onto the *PlaneY* plane (the green plane in Fig. 4(b)). The normal vector of this plane is estimated by a cross product of γ_c and $\mathbf{n}_2$ vectors ($\gamma_c \times \mathbf{n}_2$). The centroid point I_c (the red point in Fig. 4(d)) is the intersection of L_1, L_2 (two green lines in Fig. 4(c)). The radius R_c is set to the distance between I_c and p_1 on that plane. The estimated cylinder from a point cloud is illustrated in Fig. 4(d). Without loss of generality, the height of the estimated cylinder is normalized to 1. The normal vectors are computed using techniques proposed in [21]. At each point p_i, k-nearest neighbors k_n of p_i are determined within a radius r. The computation of normal vector at p_i reduces to the analysis of eigenvectors and eigenvalues of the covariance matrix C created from k_n neighbours of p_i, as given by:

$$C = \frac{1}{kn}\sum_{i=1}^{k_n}(p_i - p_{av})(p_i - p_{av})^T \ , \quad C\boldsymbol{v}_j = \lambda_j \boldsymbol{v}_j \ , \quad j \in \{0,1,2\} \tag{8}$$

where $p_{av} = \frac{1}{kn}\sum_1^{k_n} p_i$ represents the 3-D centroid of the nearest neighbors. λ_j is the j^{th} eigenvalue of the covariance matrix and v_j is the j^{th} eigenvector found by Eq. (8).

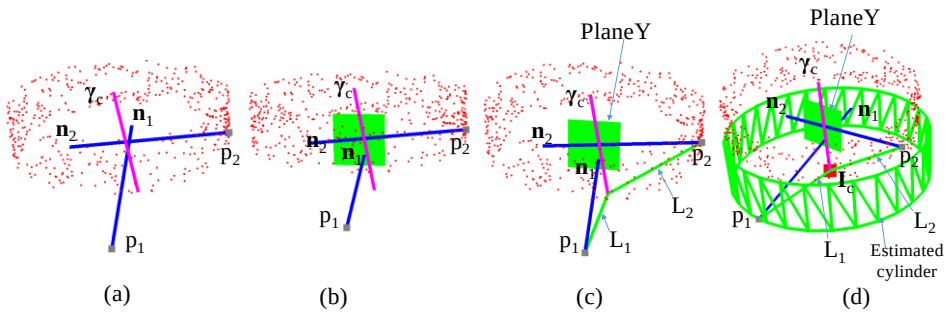


Fig. 4. Geometrical parameters of a cylindrical object. Red points are defined inlier points in the generated dataset, which has 15% inlier points. (a)-(c) Explanation of the geometrical analysis to estimate a cylindrical object. (d) Result of the estimated cylinder from a point cloud (green estimated cylinder). In this figure, the selected point p_1 is an outlier point, making the centroid point of the estimated cylinder deviated.

To deploy the geometrical constraints for cylindrical objects, let's follow illustrations in Fig. 5. In each hypothesis, a good MSS could be two within three samples p_1, p_2, and p_3, as shown in Fig. 5(a). In case of drawing two random points p_1, p_3, obviously, the first criterion is quickly satisfied because both of these samples are inliers (w_i is larger than $w_t = 0.1$). However, as shown in Fig. 4(a), the direction of the axis γ_2 is totally different from the ground-truth data, it is estimated as the cross product of $\mathbf{n}_1$, $\mathbf{n}_3$ ($\mathbf{n}_1 \times \mathbf{n}_3$). Our second criterion (or search good samples) aims to update the initial samples (for example, p_3 should be updated by p_2). To obtain this, we observe that the best case for estimating a cylinder is that normal vectors of two samples are crossed lines or intersecting together, as shown in Fig. 5(b). In the other words, $\mathbf{n}_1$ is nearly perpendicular to $\boldsymbol{n}^*_2$ where $\boldsymbol{n}^*_2$ is the projection of $\mathbf{n}_2$ onto the plane π consisting of $\mathbf{n}_1$ and the normal γ_1. This observation leads to the criterion below:

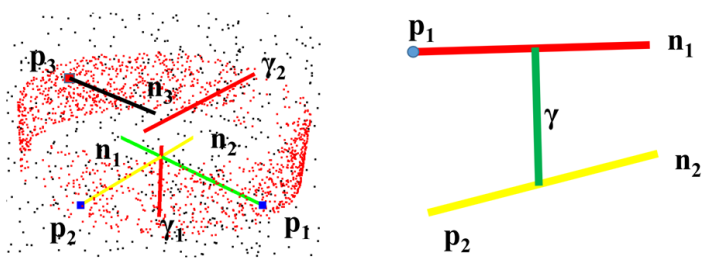


Fig. 5. Illustration of the geometrical constraints applied in GCSAC.

$$c_p = \underset{p_{2\in\{U_n\setminus p_1\}}}{\operatorname{argmin}} \{n_1.n_2^*\} \tag{9}$$

If c_p is close to 0 then $\mathbf{n}_1$ and $\boldsymbol{n_2}^*$ are orthogonal. It is noticed that in the example as shown in Fig. 4(e), the projection of $\mathbf{n}_3$ onto the plane π should be parallel to $\mathbf{n}_1$.

Therefore the dot product $\boldsymbol{n}_1 \cdot \boldsymbol{n}_2^*$ is a large scalar value.

In process of model estimation, there are some parameters that influence the estimation of the estimation model as the threshold T, the probability α that the sample is an inlier. At each iteration, a sample point is specified as an inlier whose distance to the estimated model is smaller than the threshold T. In real datasets, the threshold T is usually chosen empirically. As explained in [9], when the distribution of data is a Gaussian with zero mean and standard deviation , the threshold distance T can be estimated by a probability of $\alpha = 0.95$ that the sample is an inliers (in case of the number of minimal sample set = 2). Therefore, in the experimental evaluation, threshold distance T is set to 0.05 to ensure that only 5% of correct inliers are rejected.

3.5. Model verification using contextual constraints

During each iteration of a robust estimator (e.g., RANSAC-based algorithms), quality of the estimated model can be verified using the context's constraints. Because the cylindrical object is located on the table, the constraint can be set on the different angle between table's plane and main direction of the estimated model. We compute the deviation angle $A_d = | \gamma_c - \mathbf{n}_t |$, where $\mathbf{n}_t$ is the normal vector of the extracted table plane. At each iteration, we verify A_d with the threshold A_t, as shown in Fig. 6(c). In other words, the estimated cylinder is a wrong estimation if its main direction is not perpendicular to the table plane.

Not only verifying the estimated model by the deviation angle, the distribution of orientation of the normal vectors is an important cue. Fig. 7 illustrates two common cases. Top panel is a correct one where the cylindrical object is well fitted, as shown in the right-most illustration. Bottom panel shows a wrong estimation where the object is collected from a public dataset [22]. The main reason is that there is a bias in the distribution of the point cloud, as shown in the middle panel of Fig. 7. In the wrong estimation one, the point cloud data concentrates on a partial object (e.g., around the top-part of the object). Probability to select samples at these areas is higher than other parts. Consequently, the estimated model generated from these samples tends to be a wrong estimation. To avoid this issue, naturally, where the object lays on a table, it suggests the orientation pattern of x- axis of the normal vectors should be spread on whole directions. The statistical pattern of orientations of the normal vectors (e.g., as shown in middle panel of Fig. 7) is measured.

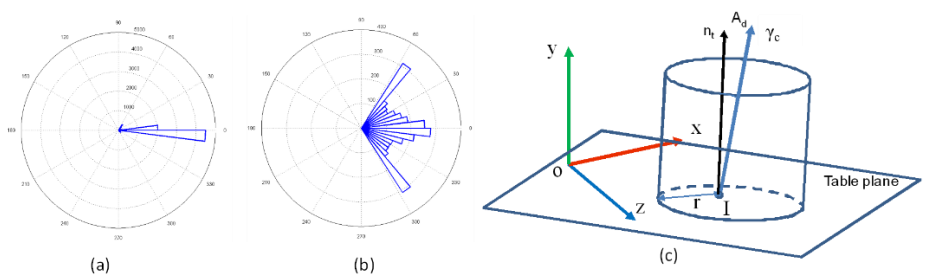


Fig. 6. (a) The histogram of deviation angles with the x-axis (1,0,0) of a real dataset in the bottom panel of Fig 7; (b) the histogram of deviation angles with the x-axis (1,0,0) of a generated dataset in

the top panel of Fig 7. (c) Illustrating of the deviation angle between the estimated cylinder's axis and the normal vector of the plane.

Figure 6(a)-(b) show the orientation distributions of x-axis (1, 0, 0)) of the normal vectors for two cases in Fig. 7, respectively. Referring to Fig. 6(a)-(b), whose normal vectors can be observed in Fig. 7, the patterns of two orientation histograms clearly indicate correct and wrong estimation, respectively. Therefore, in each iteration of GCSAC, we utilize these characteristics to validate the estimated model.

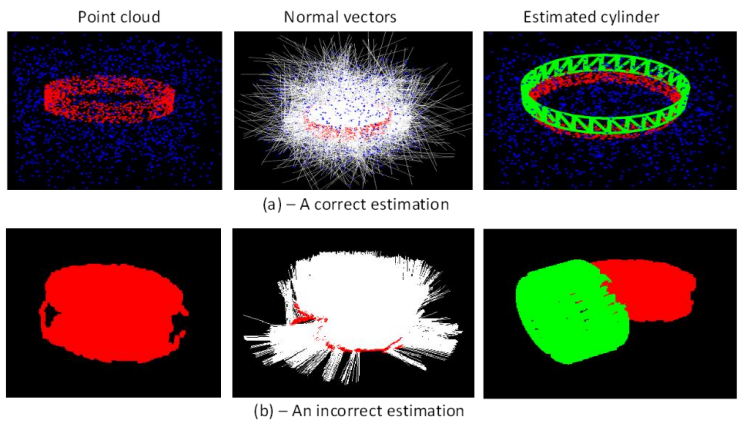


Fig. 7. Illustrations of a correct (a) and an incorrect estimation without using the verification scheme. On each sub-figure: Left panel: point cloud data; Middle panel: normal vector of each point; Right panel: the estimated model.

4. EXPERIMENTAL RESULTS

Our framework is warped by C++ programs using the PCL 1.7 library on a PC with Core i5 processor and 8G RAM. The program runs sequentially as a single thread. The performances of the proposed algorithm are evaluated in experiments for grasping cylindrical objects based on the fitting results of point clouds. In the experiments, we evaluate the proposed method on two different types of datasets. The first is a synthesized dataset and second one is real public datasets.

4.1. Impact of the searching good samples

In this section, we describe the preliminaries associated with estimating the cylindrical objects and then carry out analyses of the GCSAC with various estimation problems. It is better to use a synthesized dataset than practical one in such as evaluation. The synthesized dataset is purely artificial data, which consists of six different subsets, denoted from dC_1 to dC_6. For each subset dC_i, the inlier ratio is increased by a step of 5% from 10 to 80%. Totally, there are fifteen point clouds. They are denoted from dS_1 to dS_{15}. A point cloud dS_i consists of three thousand points. An inlier data point (x_i,y_i,z_i) of dS_i lying on cylinders curved-surface is generated as follows: $x_i = cos(\theta_i), z_i = sin(\theta_i), y_i$ is randomly

selected in [0, 1], θ_i is in a range of [0, 2π]. In the synthesized point clouds, outliers are generated randomly as both uniform and normal distributions. The major differences between a dC_i and dC_j could be the main axis's orientation, the deviation value σ of the normal distribution for generating outliers/inliers; or the spatial distribution of inliers. Fig. 8 illustrates the synthesized data of dC_1, dC_2, dC_3 whose inlier ratio equals to 50%.

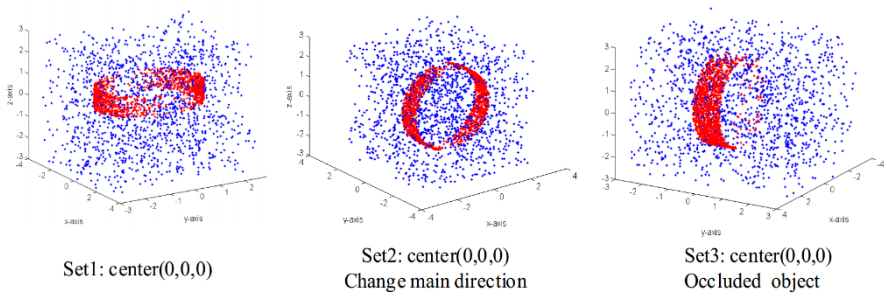


Fig. 8. Illustrations of dC_1, dC_2, dC_3 point clouds of the first dataset in case of 50% inlier ratio. The red/blue points are inliers/outliers, respectively.

In the evaluations, the proposed GCSAC is compared with the original MLESAC algorithms [4]. First, we record residual errors (or total distance from outliers to the estimated model) and inlier scale estimation at each iteration. Fig. 9(a) shows the residual errors at different inlier rates. At each inlier rate, the residual error is calculated as total distances from outliers to the best estimated model. It should be noticed that because some outliers may be scatted within a range $[-T, T]$ (the threshold distance T is to indicate an inlier sample), that means the total residual error is smaller than the actual distance of the outliers from groundtruth data. Obviously, GCSAC and MLESAC have equal performance. More specifically, Fig. 9(b) shows the relative error distance of the total outliers and ideal case. As shown, GCSAC achieves even better result with lower inlier scale. However, GCSAC requires a limited number of iterations, as shown in Fig.10. It is noticed that both estimators (GCSAC and MLESAC) use scheme of adaptive RANSAC to update the number of sample selections when a better model is achieved. Although the number of iterations does not directly indicate the computational time of GCSAC (versus MLESAC). Let's consider a specific case below for further analyzing the GCSAC's implementation.

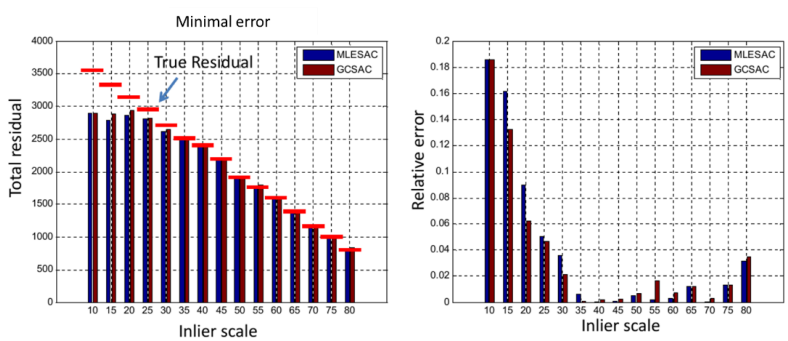


Fig. 9. Comparisons of GCSAC and MLESAC algorithms. (a) Total residual errors of the estimated model and ideal case. (b) Relative errors of the estimated model and idea case.

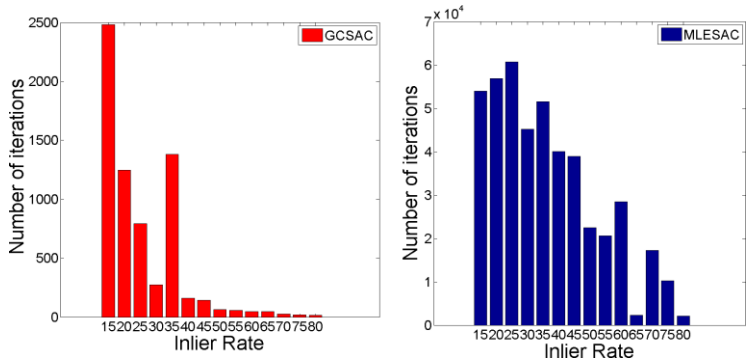


Fig. 10. The number of iterations of GCSAC and MLESAC on the synthesized dataset.

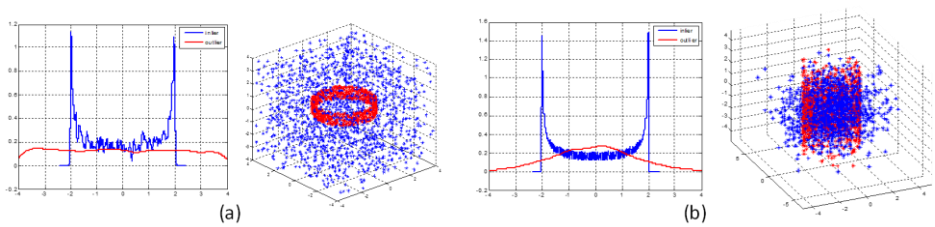


Fig. 11. Decomposition of residual density distribution: inlier (blue) and outlier (red) density distributions of a synthesized point cloud with 50 inliers. (a) Noises are added by an uniform distribution. (b) Noises are added by a Gaussian distribution ($\mu = 0$, $\sigma = 1.5$). In each subfigure, left-panel shows the distribution of an axis (e.g., x-axis), right-panel shows the corresponding point cloud.

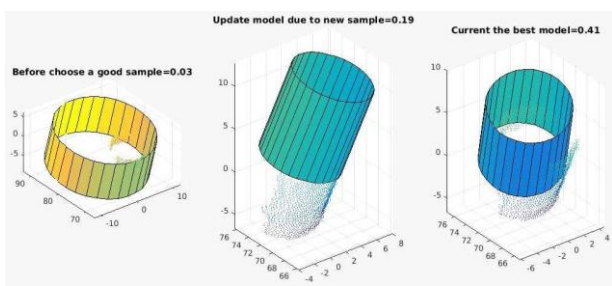


Fig. 12. An illustration of GCSAC's at a k^{th} iteration to estimate a coffee mug in the second dataset. Left: the fitting result with a random MSS. Middle: the fitting result where the random samples are updated thank to the geometrical constrains. Right: the current best model.

Let's consider the specific synthesized dataset that consists of 50% inlier. The distributions of inliers and outliers of this dataset are decomposed as shown in Fig.11(a) in case of uniform distribution and Fig.11(b) in case of normal distribution. In GCSAC's implementation, at the iteration k, two random samples (p_1, p_2) are chosen for estimating a cylinder as MLESAC. p_1 is kept and p_2 is found based on the p_1 following the geometrical constraint as Eq. 9. When this case appears, as shown in Fig. 12, combining p_1 and p_2 generates a good cylinder which has high inlier ratio and the residual error is near to that

calculated from the ground-truth model. Thanks to the high inlier, the iteration k is reduced as Eq. 7. GCSAC therefore converges faster. Figure 13 shows results of the estimated model with 50% inlier synthesized dataset. Consequently, the effectiveness of "good sampling" strategy, as proposed by GCSAC, is confirmed.

4.2. Descriptions of real datasets and evaluation measurements

To compare the performances of GCSAC with MSLESAC [4] in real scenarios, we utilize three datasets that included the public and our own preparations. All of them are captured in practical environments, consist of many noises, and are challenging with different sizes. The first dataset [22] contains calibrated RGB-D data collected by a Microsft Kinect v1 of 111 indoor scenes. To adapt to this study, only scenes that consist of cylindrical structures are manually selected. Some instances are illustrated in Fig. 14(a)-(b). The second dataset is published in [23]. It consists of 14 scenes containing furniture (chair, coffee table, sofa, table) and a set of the cylinder-like objects such as bowls, cups, coffee mugs, and soda cans. For this dataset, we only selected the relevant scene (e.g., 2^{th}, 4^{th}, 9^{th} scenes so on) where the cylinder-like objects appear. Each scene has around 800 frames, each frame consists of more than one cylindrical objects on the table. In this dataset, the radius of coffee mugs, bowls, soda cans are 3.75cm, 5cm, 2.5cm, respectively. Their heights are 10cm, 7cm, 10cm. The ground truth of cylindrical objects in these datasets is manually prepared using a visualization tool of PCL library.

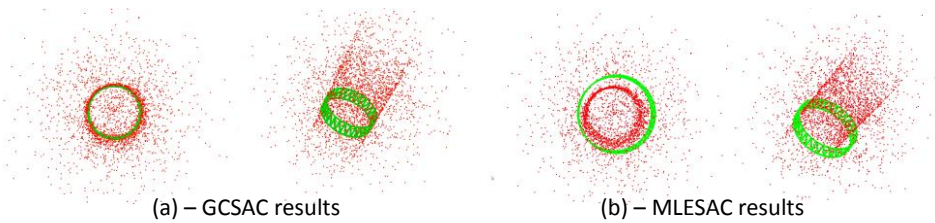


Fig. 13. The best estimated model using GCSAC (a) and MLESAC (b) with 50% inlier from a synthesized point cloud. In each sub-figure, two different view-points are given.

Figure 14(c) shows a ground-truth preparation example that a line connecting two selected points on the top of the interested object specifies a cylinder. The third dataset is collected by ourself in indoor environments (e.g., cafeteria, showroom) where the cylindrical objects (e.g., coffee cups, bottles) are on a table. There are six types of the cylinder-like objects as shown in Fig. 15. Their radii range from 3.5cm to 4.5cm with various heights (from 6.0 cm to 20 cm). A Microsoft Kinect v1 is mounted on the chest of a person who moves around a table. The dataset consists of 8 scenarios, each includes about 200 frames. In addition, we put some contaminated objects such as boxes ($10.0cm \times 30.0cm$) besides the cylindrical objects. This dataset is built to adapt to the context of practical application supporting visually impaired people finding an interested object.

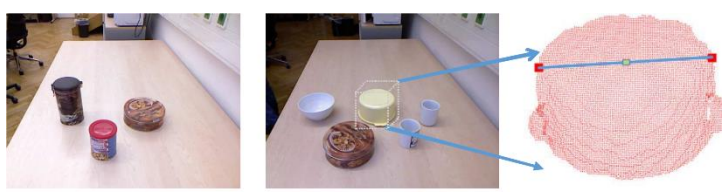


Fig. 14. Some examples of the cylindrical objects [22] collected from the first dataset.

Fig. 15. Illustration of six types of cylindrical objects in the third dataset.

Fig. 16. Result of table plane detection in a pre-processing step using the methods in [24, 21]. (a) A RGB image of current scene; (b) The detected table plane is marked in green points. (c) The point clouds above the table plane are located and marked in red.

It is noticed that table plane in each scene is detected in a pre-processing step. Fig. 16 illustrates the detection result in which the table plane is marked in green points (Fig. 16(b)). The point clouds data above the table plane remain for further fitting, as shown in Fig. 16(c).

To evaluate the performance of the proposed method, some features of the cylindrical objects such as radius R, and position (or main axis direction γ) can be used. We denote a ground-truth and estimated cylindrical object C_t and C_e, respectively. It is noticed that the height of a cylinder object is normally calculated in an additional step. For example, it is determined by the maximal distance between two projected points in [19]. In this study, the height is set to 1. For quantitative evaluation, we used three following measurements:

- E_a (*degree*) the different angle between the main direction of the estimated cylinder γ_c and the normal vector of table plane z_t.

$$E_a = |\gamma_c - z_t| \tag{10}$$

- E_r (%) the relative error between the radius of the estimated cylinder (R_e) and the ground truth one (R_g).

$$E_r = \frac{|R_e - R_g|}{R_g} \times 100\% \tag{11}$$

- t_p the processing time, measured in milliseconds (ms).

In these evaluations, the smaller indexes (e.g., E_a, E_r, t_p) are, the better method is. To evaluate the role of contextual constraints (as described in Section 3.5), the quantitative

measurements are determined with and without using the proposed constraints. In experiments, we fixed the threshold of estimators T = 0.01 (or 1cm), the weight w_t = 0.1 and A_t = 20 degrees. T is a distance threshold to set a data point to be an inlier or outlier. For fair evaluation, T is set equally for both fitting methods.

4.3. Evaluation results on real datasets

Figure 17 shows some fitting results from the second and the third dataset. For comparative evaluations, Table 1 compares the performances of the proposed method (GCSAC) and MLESAC. In this table, E_a, E_r, t_p are averaged on whole fitting results of three datasets. Compared with MLESAC, the objects estimated by GCSAC algorithm are highly accurate. The most differences between GCSAC and MLESAC can be observed from the fitting results for the first and the second datasets. While MLESAC always obtains E_a from 45^o to 47^o, the GCSAC has lower E_a from 10^o for the first dataset to 2^o for the third dataset. The computational time is clearly different between GCSAC and MLESAC. Comparing three datasets, the first dataset has the highest error of the estimated radii: MLESAC is 92.85%, GCSAC is 81.01%. These errors come from missing data issues or many noises appear in green jars, yellow bottles, and pink bottles, as illustrated in Fig. 18.

(a) (b)

Fig. 17. (a)-(b) Some examples of fitting results from the second and the third dataset. In these scenes, there is more than one cylindrical object. They are marked in red, green, blue and yellow, so on. The estimated parameters include radius, position (a center of the cylinder), main axis direction. The height can be computed using normalization in y-value of the estimated object.

Table 1. Average results of the evaluation measurements using GCSAC and MLESAC on three datasets. The fitting procedures were repeated 50 times for statistical evaluations.

Dataset	Method	Error measurements		
		$E_a(deg.)$	E_r (%)	$t_p(ms)$
First dataset	*MLESAC*	46.47	92.85	18.10
	GCSAC	**36.17**	**81.01**	**13.51**
Second dataset	*MLESAC*	47.56	50.78	25.89
	GCSAC	**40.68**	**38.29**	**18.38**
Third dataset	*MLESAC*	45.32	48.48	22.75
	GCSAC	**43.06**	**46.9**	**17.14**

In these experiments, E_a and E_r are still large as reported in Table 1, even with the fitting results using GCSAC. This issue also can be observed in Fig. 18. Radii of the blue (Fig.

18(a)) and green (Fig. 18(b)) objects are much larger than the groundtruth one. It is noticed that the evaluation results reported in Table 1 come from GCSAC without using the contextual constraints to verify the estimated model. The effectiveness of contextual constraints is shown in Fig. 19. Obviously, by using the context's constraints, the estimated objects could be eliminated when there is a large angle error. The deviation angle is reduced from about 40^o to 12^o. Hereby, the full model of objects is more accurate for grasping task. This verification step suggests a solution to resolve estimating inlier threshold T which is a common issue of the RANSAC-based algorithms.

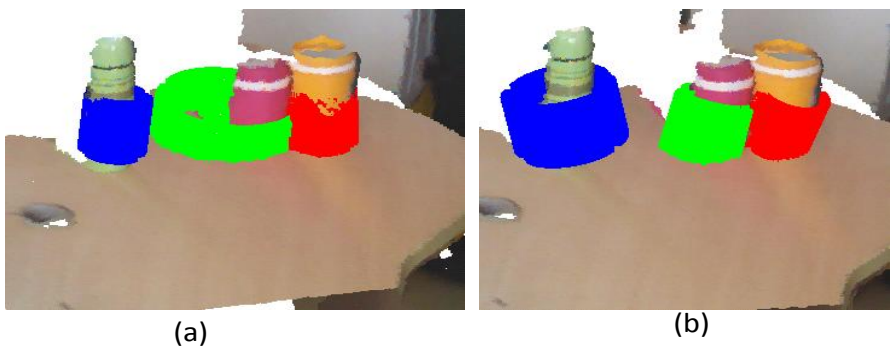


Fig. 18. (a) The estimated cylindrical object (in green) has relative error of radius E_r =111.08%; (b) The estimated cylindrical object (in blue) has the relative error of radius E_r =165.92%.

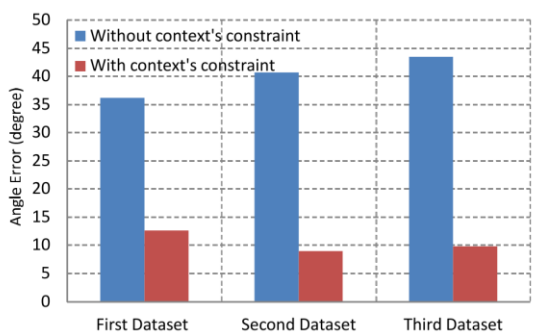


Fig. 19. Angle errors E_a of the fitting results using GCSAC with and without using contextual constraints.

4.4. Discussion

Beyond proposing a new sampling strategy for a robust estimator, our final aim is to develop the object-finding-aided system for visually impaired people. The entire procedure of the proposed framework consists of RGB and depth data collection from Kinect, table detection, and objects fitting. By experiment, all of these steps require 1.04 second per frame. In these procedures, we do not down-sample the data. Fig. 20 shows snap-shots from one minute of video, taken from common scene in an indoor environment of the third dataset. The completed video and relevant scenes are available at *http://mica.edu.vn/perso/Le-Van-Hung/videodemo/index.html*.

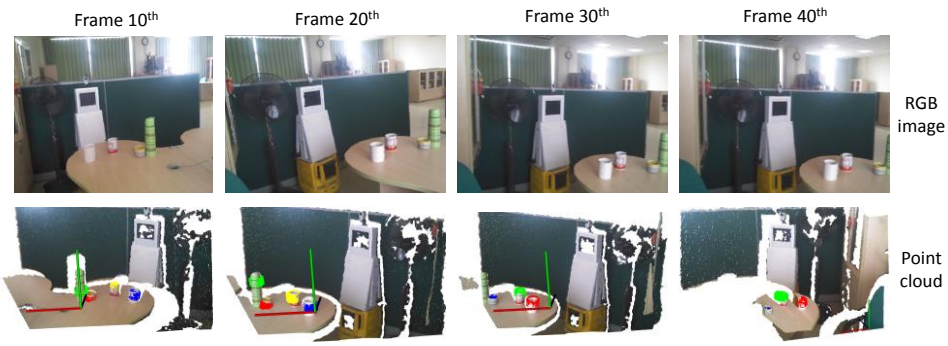


Fig. 20. Fitting results extracted from of one video of the first dataset.

The proposed method also successfully locates multiple objects in scenes that are more complex (e.g., there are four cylindrical objects on a table). Consequently, the proposed method could be feasible for deploying a completed system supporting visually impaired people in their daily activities.

The fact the final goal could be formed as 3-D object recognition task (e.g., [11, 12, 25, 26]. This research field has been widely attempted in computer vision and robotics communities. Most of 3-D recognition techniques tend to address challenging issues such as occlusion, free-from styles, and unconstrained scene. To do this, a model scene matching (e.g., using point-pair feature, or geometrical consistent of local points) always is required in these approaches. Different from these works, the proposed method tends to use geometrical analysis of an interested object rather than using a *prior* model (or template) of object (in order to match between object and scene). However, the proposed method is suitable to some objects associated with geometrical analysis such as primitive shapes (spheres, cylinder, boxes, cone, so on) but not appropriate for free-form style objects. The detailed comparisons between a geometrical-based technique (e.g., the proposed method) and matching-based approaches for 3-D object recognition are out of scope. It suggests us future research directions in which critical factors such as computational time, accuracy of the estimated model, and cost for preparing training data of these types of the approaches will be comprehensively compared and examined.

5. CONCLUSIONS

In this paper, we proposed a new framework for fitting the cylindrical objects in the scene. We proposed to use some geometrical and contextual constraints for deploying the fitting algorithms. Not only proposed GCSAC, the context's constraints used for verifying the estimated model were proposed. In the experimental results, GCSAC is evaluated by quality of the estimating cylinders with various size in different practical scenarios and is compared with a common robust estimator (e.g., MLESAC). The performances of the proposed robust estimator GCSAC were confirmed. It could estimate cylindrical objects from point clouds that have been contaminated by noise and outliers. The average processing time of our proposed method is acceptable to deploy a real application.

Therefore, it suggests us deploying the real application as aided-service for impaired/blind people. The application helps to query common objects in the kitchen or cafeteria. In the future, we continue to extend GCSAC for fitting other primitive shapes such as spherical, conical objects.

ACKNOWLEDGMENT

This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2016-LN-27.

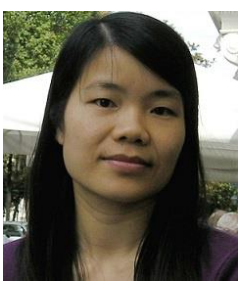
REFERENCES

1. R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detec-tion," Computer Graphics Forum, Vol. 26, no. 2, 2007, pp. 214–226.
2. M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysisand automated cartography," Communica-tions of the ACM, Vol. 24, no. 6, 1981, pp. 381–395.
3. O. Chum and J. Matas, "Matching with prosac progressive sample consensus," in IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 220–226.
4. P. H. S. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," Computer Vision and Image Understanding, Vol. 78, no. 1, 2000, pp. 138–156.
5. R. Raguram, J. Frahm, and M. Pollefeys, "A comparative analysis of ransac tech-niques leading to adaptive real-time random sample consensus," in European Con-ference on Computer Vision, 2008, pp. 500–513.
6. S. Choi, T. Kim, and W. Yu, "Performance evaluation of ransac family," in British Machine Vision Conference, 2009, pp. 1–12.
7. O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac." in DAGM-Symposium, ser. LNCS, Vol. 2781. Springer, 2003, pp. 236–243.
8. K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized ransac," in British Machine Vision Conference, 2012, pp. 3–7.
9. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
10. C. S. Chen, Y. P. Hung, and J. B. Ceng, "Ransac-based darces: a new approach to fast automatic registration of partially overlapping range images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. Volume: 21, 1999, pp. pp 1229 – 1234.
11. B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model Globally, Match Locally: Efficient and Robust 3D Object Recognition," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 998–1005.
12. B. Tolga and I. Slobodan, "Point Pair Features Based Object Detection and Pose Estimation Revisited," in International Conference on 3D Vision (3DV), 2015, pp. pp 527–535.

13. R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J. M. Frahm, "Usac: A universal framework for random sample consensus," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, no. 8, Aug 2013, pp. 2022–2038.
14. M. Kohei, U. Yusuke, S. Shigeyuki, and S. Sato, "Geometric verification using semi-2d constraints for 3d object retrieval," in International Conference on Pattern Recog-nition, 2016, pp. 2339–2344.
15. K. Alhamzi and M. Elmogy, "3d object recognition based on image features : A survey," International Journal of Computer and Information Technology, Vol. 03, no. 03, 2014, pp. 651–660.
16. K. Duncan, S. Sarkar, R. Alqasemi, and R. Dubey, "Multiscale superquadric fitting for efficient shape and pose recovery of unknown objects," in IEEE International Conference on Robotics and Automation, 2013.
17. G. Osselman, B. Gorte, G. Sithole, and T. Rabbani, "Recognising structure in laser scanner point clouds," in International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 2004, p. 33-38.
18. T. T. Tran, V. T. Cao, and D. Laurendeau, "Extraction of cylinders and estimation of their parameters from point clouds," Computers and Graphics, Vol. 46, 2015, pp. 345–357.
19. S. Garcia, "Fitting primitive shapes to point clouds for robotic grasping," Master Thesis, 2009.
20. V. H. Le, V. Michiel, H. Vu, T. T. Nguyen, T. L. Le, T. H. Tran, Q. H. Luong, V. Peter, and W. Philips, "Real-Time Table Plane Detection Using Accelerometer Information And Organized Point Cloud Data From Kinect Sensor," Journal of Computer Science and Cybernetics, 2016, pp. pp 243–258.
21. S. Dirk, R. B. Rusu, and S. Behnke, "Real-Time Plane Segmentation Using RGB-D Cameras," in LNCS, 2011, pp. 306–317.
22. A. Richtsfeld, T. Mrwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 4791–4796.
23. L. Kevin, B. Liefeng, and F. Dieter, "Unsupervised feature learning for 3d scene labeling," in IEEE International Conference on Robotics and Automation, 2014.
24. V. H. Le, T. L. Le, H. Vu, T. T. Nguyen, T. H. Tran, T. C. Dao, and H. Q. Nguyen, "Geometry-based 3-d object fitting and localization in grasping aid for visually im-paired," in IEEE Seventh International Conference on Communications and Elec-tronics, 2016.
25. D. Andreas, K. Rigas, M. Sotiris, and K. Tae-Kyun, "Recovering 6D Object Pose and Predicting Next-Best-View in the Crowd," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 998–1005.
26. G. B. Andres, K. Lilita, and K. Dirk, "Rotational Subgroup Voting and Pose Cluster-ing for Robust 3D Object Recognition," in Proceeding of International Conference on Computer Vision (ICCV 2017), 2017.

Hai Vu received B.E. degree in Electronic and Telecommunications in 1999 and M.E. in Information Processing and Communication in 2002, both from Hanoi University of Science and Technology (HUST). He received Ph.D. in Computer Science from Osaka University, 2009. He has been a lecturer at MICA International Research Institute since 2012. He is interested in computer vision techniques for medical imaging, HCI, robotics.

Van-Hung Le received his M.Sc. at Faculty Information Technology- Hanoi National University of Education (2013). He is now an PhD at International Research Institute MICA HUST- CNRS/UMI - 2954 - INP Grenoble. His research interests include Computer vision, RANSAC and RANSAC variation and 3D object detection, recognition.



Thuy Thi Nguyen received her BSc. degree in Math & Informatics in 1994, MSc. in Information Technology from HUST in 2002, and PhD. in Computer science from Graz University of Technology, Austria in 2009. She has been a lecturer at Faculty of Information Technology, Vietnam National University of Agriculture since 1998, head of department of Computer Science since 2011. Her research interest includes object recognition, visual learning, video understanding, statistical methods for computer vision and machine learning

Thi-Lan Le graduated in Information Technology HUST, Vietnam. She obtained MS degree in Signal Processing and Communication from HUST, Vietnam. In 2009, she received her Ph.D. degree at INRIA Sophia Antipolis, France in video retrieval. She is currently lecturer/researcher at Computer Vision Department, HUST, Vietnam. Her research interests include computer vision, content-based indexing and retrieval, video understanding and human-robot interaction.

Thanh-Hai Tran graduated in Information Technology from Hanoi University of Science and Technology in 2001. She received ME. then Ph.D In Imagery Vision Robotic from INPG in 2002 and 2006 respectively. Since 2009, she is lecturer/researcher at Computer Vision group, International Institute MICA, Hanoi University of Science and Technology. Her main research interests are visual object recognition, video understanding, human-robot interaction and text detection for applications in Computer Vision.