Fitting Cylindrical Objects in 3-D Point Cloud Using the Context and Geometrical Constraints

Hai Vu¹, Van-Hung Le¹,², Thi Thuy Nguyen³, Thi-Lan Le¹ and Thanh-Hai Tran¹

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 Agriculture, Vietnam

In this paper, we propose a frame-work for fitting cylindrical objects toward deploying an object-finding-aided system for visually impaired people. The proposed frame-work consists of a RANSAC-based algorithm and a model verification scheme. The proposed robust estimator named GCSAC (Geometrical Constraint SAmple Consensus) avoiding expensive computation of the RANSAC-based algorithms due to its random drawing of samples. To do this, GCSAC utilizes the geometrical constraints for selecting good samples. These constraints are delivered 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 context constraints which could be delivered from a certain scene such as object standing on a table plane, size of an object, and so on. In experimental result, GCSAC’s implementations are analysed for various estimation problem on the synthesized dataset. The comparisons between GCSAC and MLESAC algorithm are implemented on with three public datasets in terms of accuracy of the estimated model and 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, cone, from 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]. A Random Sample Consensus (RANSAC) [2] and its paradigm attempt to extract as good as possible shape parameters which are objected either heavy noise in the data or processing time constraints. Originally, a RANSAC algorithm consists of hypotheses based on drawing randomly 3-D points from an input data set. Although its variants such as PROSAC algorithms [3] proposed a so-called guided sample schemes, they still need to further investigate to efficiently adapt to real scenarios. The fact that many applications in practice, a priori knowledge is often available, and this can be used to generate better hypotheses. As results, a better estimated model is achievable. In this paper, we propose a new RANSAC-based algorithm, named Geometrical Constraint SAmple Consensus - GCSAC. The proposed algorithm is inspired by guiding the minimal subset using normal constraints of geometric models.

Received july 8, 2017
Query-based: “Where is coffee cup (or box, bottle)?”

Fig. 1. Finding a query-based object to support the visually impaired people in a cafeteria room.

and the context of scene. We demonstrate GCSAC for fitting cylindrical objects in a real application which supports visually impaired people.

Let’s consider a real scenario in common daily activities of the visually impaired people. They come to a cafeteria room then give a query "where is a coffee cup?", as shown in Fig.1. To estimate position of the coffee cup, we formulate this problem as fitting a primitive shape (e.g., a cylinder object) from a 3-D point cloud data collected by a Kinect sensor. The prior knowledge can be observed from the current scene. For instance, the coffee cup should stand on a table. Other contextual constraints could be walls in the scene to be perpendicular with the table plane; size of the table plane is limited. These constraints derive many cues for deploying the RANSAC-based approaches. Specially, we attempt to search qualified (or good) samples that ensure geometrical constraints of the interested object(s). In addition, the context constrains help to verify the estimated model in a post-processing step.

In this paper, we first describe the GCSAC for fitting an query-based object like a cylindrical model. While a RANSAC paradigm randomly draws 3-D points from an input data set without any prior assumption on the data, at each hypothesis of GCSAC, a searching process aims at finding good samples based on the constraints of an estimated model. To perform the search for good samples, we define two criteria: (1) The selected samples must ensure being consistent with the estimated model via a roughly inlier ratio evaluation; (2) The samples must satisfy explicit geometrical constraints of the interested objects (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-Likelihood criteria as defined by MLESAC algorithm. Moreover, the estimated model is verified using contextual constraints. In the experimental evaluations, the proposed algorithm is compared with original MLESAC [4] using three realistic datasets in terms of both quality of the estimated model and computational time. These datasets consist of cylindrical objects which 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 a primitive shapes parameters, particularly, focusing on estimations of cylindrical objects.

For the first topic, because the original RANSAC is very general with a straightforward implementation, it 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<sup>+</sup>-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 is immediately terminated when a smaller number of iterations has been obtained. 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, Chu-Song 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] and Tolga et al. [12] propose a framework that creates a global model description based on point pair features. Matching between a model and scene utilizes an efficient Hough-like voting scheme on a reduced pose parameter space. Recently, USAC [13] introduces 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 does not really deploy an estimator for primitive shape(s) from a point cloud. A recent work [14] proposes 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 achieving the best model as soon as possible. Therefore, the number of iterations is significantly reduced thanks to the results of the search 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] formulates a cylinder using three parameters such as radius \( r \), height \( h \), and the axis of the cylinder \( \gamma \). [11] defines a cylinder through two samples and their normal vectors. In
Fig. 2. The general framework for fitting the cylindrical objects using the geometrical constraints.

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

3.1 A general frame-work

In the context of developing an aided-systems for visually impaired people (as shown in Fig [1]), a general frame-work is presented in the Fig. 2. While the previous work [20] achieve a high accuracy and real time performance on the table plane detection by utilizing a combination of depth and acceleration features coming from a Microsoft Kinect sensor. In this paper, we describe remaining steps of the frame-work which are marked in red blocks in the Fig. 2.

3.2 Separating the table plane and interested objects

Given a normal vector \( \mathbf{n} \) 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 \( \mathbf{n}_t \). The transformation between two coordinate systems is described by rotation and translation matrices. The rotation matrix are:

\[
R_x(\alpha) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha \\
0 & -\sin \alpha & \cos \alpha
\end{bmatrix}
\]

(1)

\[
R_y(\beta) = \begin{bmatrix}
\cos \beta & 0 & -\sin \beta \\
0 & 1 & 0 \\
\sin \beta & 0 & \cos \beta
\end{bmatrix}
\]

(2)

\[
R_z(\gamma) = \begin{bmatrix}
\cos \gamma & \sin \gamma & 0 \\
-\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

(3)

where \( \alpha, \beta, \gamma \) are rotation angles in x, y, z axis.

From Eq. 1, Eq. 2 and Eq. 3, we have the rotation matrix:

\[
R = R_z(\gamma)R_y(\beta)R_x(\alpha)
\]

(4)

In this context, the transformation is defined by a rotation and a translation operator. The rotations in x-axis, z-axis are defined by angle \( \alpha, \gamma \):

\[
\alpha = \arcsin \frac{b}{\sqrt{b^2 + c^2}} \quad \gamma = \arcsin \frac{c}{\sqrt{c^2 + a^2}}
\]

(5)
Fitting Cylindrical Objects in 3-D Point Cloud...

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 \( O_t(x_t, y_t, z_t) \), that the normal vector \( n_t \) of a table plane is parallel to the y-axis.

where \((a, b, c)\) are components of the normal vector \( n_t \) of table plane. We perform a translation \( d \) in y-axis by a term:

\[
d = |y_t|
\]  

(6)

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 belonging to the table plane as shown in Fig. 3. They will be preserved based on an assumption that the query objects are always laid 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 not be small enough. In the next step, the point clouds of a cylindrical object are fitted using the proposed GCSAC algorithms.

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 selection is updated by:

\[
K = \frac{\log(1 - p)}{\log(1 - w^m)}
\]  

(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 to
Algorithm 1: GCSAC’s implementation for fitting a cylindrical object from the point cloud

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

Algorithm:
1. Step 1: initialization
2. Step 2: While ($k < K$)
   3. \{\ 
      4. \quad k++;
      5. \quad \text{Drawing randomly two points } P = \{p_1, p_2\} \text{ from } U_n;
      6. \quad U^*_n = \emptyset;
      7. \quad \text{if } (U^*_n \neq \emptyset) \text{ estimate model } M_k \text{ from } P \text{ else goto 2.1
      8. \quad \text{Compute } w_k$
      9. \quad \text{if } (w_k \geq w_l) \text{ and } (w_k > w_m)\{
         10. \quad \text{Search } p^*_k \text{ by (9)};
         11. \quad \text{Update } U^*_n = \{p_1, p^*_2\};
         12. \quad w_m = w_k;$
      13. \} \text{ 2.6 Re-estimate } M_k \text{ from } U^*_n
      14. \quad \text{Compute } A_d = (\gamma_c, n_t)$
      15. \quad \text{if } (A_d < A_t) \text{ compute } -L \text{ else goto 2.1
      16. \quad \text{if } (-L < L_t)\{
         17. \quad \text{choose the best model } M_b;
         18. \quad \text{re-compute } K;$
      19. \} \text{ 2.9 else goto 2.1
   \}

set 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.

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. At each iteration, we assume that the worst case of inlier ratio $w_i = 0.1 (10\%)$ is determined, to initialize the stack $U^*_n$, where $U^*_n$ is used to store $m - 1$ kept points and its inlier ratio $w_i$. A consensus set therefore containing more than 10\% of the data is easily found. A model is estimated from $m$ random samples. As estimating a cylinder is $m = 2 [1]$. After that, $U^*_n$ is reset. $m$ samples and the inlier ratio $w_i$ of the estimated model is stored into $U^*_n$ if $w_i$ is equal to or greater than $w_l$. And 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 which satisfy the geometrical principles of a primitive shape are explained in Section 3.4. If none of iterations find out that satisfies $w_i \geq w_l$, the estimation algorithm degrades to the original MLESAC. The inlier ratio of a iteration depends on the threshold $T$, which chooses an optimal $T$ value is out of scope of this research.

3.4 Geometrical analyses for qualifying good samples

A cylinder is determined by some 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 $\gamma_c$; and its radius
an outlier point, making the centroid variance matrix in Fig. 5. In each hypothesis, a good MSS could be two within three samples as shown in Fig. 5(b). In the other words, a cylinder is that normal vectors of two samples are crossed lines or intersecting together,

\[ \gamma = n_1 \times n_2. \]

To estimate a centroid point \( C \) and \( 2 \times 2 \) covariance matrix, and \( \lambda_v \) is the first eigenvalue of the covariance matrix, and \( v_j \) is the \( j \)-th eigenvector found by Eq. (8).

To deploy the geometrical constraints for cylindrical objects, let’s follow illustrations in Fig. 4. In each hypothesis, a good MSS could be two within three samples \( p_1, p_2, \) and \( p_3 \), as shown in Fig. 4(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 \( w_i = 0.1 \)). However, as shown in Fig. 4(a), the direction of the axis \( p_2 \) is totally different from the ground-truth data, it is estimated as the cross product of \( n_1, n_3 \) (\( n_1 \times n_3 \)). Our second criteria (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. 4(b). In the other words, \( n_1 \) is nearly perpendicular to \( n_2 \). Let \( L_1, L_2, L_3 \) be 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.

\[ R_c. \] For geometrical analyses of a cylinder object, we adopted the analysis given by [11]. 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 \((n_1, n_2)\) (blue lines in Fig. 4(a)). Let \( \gamma \) is the main axis direction (pink line in Fig. 4(a)) of the cylinder. It is estimated by \( \gamma = n_1 \times n_2 \). To estimate a centroid point \( C \), we project the two parametric lines \( L_1 = p_1 + t n_1 \) and \( L_2 = p_2 + t n_2 \) along the axis onto the PlaneY plane (see a green plane in Fig. 4(b)). The normal vector of this plane is estimated by a cross product of \( \gamma \) and \( n_2 \) vectors \((\gamma \times n_2)\). The centroid point \( C \) is set by the distance between \( I \) and \( p_1 \) on that plane. The estimated cylinder from a point cloud is illustrated in Fig. 4(d). 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 \( r \). Computation of the normal vector at \( p_i \) is reduced to the analysis of eigenvectors and eigenvalues of the covariance matrix \( C \) created from \( k_n \), the neighbours of \( p_i \), as given by:

\[ C = \frac{1}{kn} \sum_{i=1}^{kn} (p_i - p_{av})(p_i - p_{av})^T, \quad CV_j = \lambda_j v_j, \quad j \in \{0, 1, 2\}; \]

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

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 \( w_i = 0.1 \)). However, as shown in Fig. 5(a), the direction of the axis \( p_2 \) is totally different from the ground-truth data, it is estimated as the cross product of \( n_1, n_3 \) (\( n_1 \times n_3 \)). Our second criteria (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, \( n_1 \) is nearly perpendicular to \( n_2 \). Let \( L_1, L_2, L_3 \) be 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.
leads to the criteria below:

\[ c_p = \arg\min_{p_2 \in \{U \setminus p_1\}} \{ n_1 \cdot n_2 \} \]  

(9)

If \( c_p \) is close to 0 then \( n_1 \) and \( n_2 \) are orthogonal. It is noticed that in the example as shown in Fig. 4(c), projection of \( n_3 \) onto plane \( \pi \) should be parallel to \( n_1 \). Therefore the dot product \( n_1 \cdot n_3 \) is a large scalar value.

In process of model estimation, there are some parameters that influence the estimation of the estimation model as: threshold \( T \), a probability \( \alpha \) 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 a threshold \( T \). In real datasets, this distance threshold \( T \) is usually chosen empirically. As explained in [9], when the distribution of the data is a Gaussian with zero mean and standard deviation , the threshold distance \( T \) can be estimated by for a probability of \( \alpha = 0.95 \) that the sample is an inlier (in case of the number of minimal sample set = 2). Therefore, in the experimental evaluation, threshold distance \( T \) is equal 0.05. It ensures that an inlier will only be incorrectly rejected 5% of the time.

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 \( A_d = |\gamma - n_t| \), where \( n_t \) 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. 5(c). In other word, the estimated cylinder is a wrong estimation if its main direction is not perpendicular with 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 rightmost illustration. Bottom panel is 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 is concentrated on a partial object (e.g., around the top-part of the object). Probability selecting samples at these area is higher than other parts. Consequently, the estimated model that is 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 panels of Fig. 7) is measured.
4. EXPERIMENTAL RESULTS

Our framework is warped by C++ programs using a 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 the evaluations. The synthesized dataset is purely artificial data which consists of six different subsets, denoted from dC₁ to dC₆. For each subset dCᵢ, inlier ratio is increased by a step of 5% from 10 to 80%. Therefore, there are fifteen point clouds. They are denoted from dS₁ to dS₁₅. A point cloud dSᵢ consists of three thousand points. An inlier data point (xᵢ, yᵢ, zᵢ) of dSᵢ lying on cylinders curved-surface is generated as follows: xᵢ = cos(θᵢ), zᵢ = sin(θᵢ), yᵢ is randomly selected in [0, 1], θᵢ is selected from [0, 2π]. In the synthesized point clouds, outliers are generated randomly as both uniform and normal distributions. The major differences between a dCᵢ and dCⱼ 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 dC₁, dC₂, dC₃ whose inlier ratio equals to 50%.

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

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.
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 scattered within a range $[-T, T]$ (the threshold distance $T$ is to indicate an inlier sample), that means total residual error calculated are smaller than the actual distance of the outliers that come from ideal data. Obviously, GCSAC and MLESAC have equal performance. More specifically, Fig. 9(b) shows the relative distance error 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 should be noticed that both estimators (GCSAC and MLESAC) using scheme of adaptive RANSAC that updates the number of sample selections when a better model is achieved. Although the number of iterations does not directly indicate computational time of GCSAC (versus MLESAC), Let’s consider a specific case below for further analyzing GCSAC’s implementations.

Let’s consider the specific synthesized dataset that consists of 50% inlier. The distri-
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 a 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 due to applying the geometrical constrains. Right: the current best model.

Distributions of inlier and outlier of this dataset are decomposed as shown in Fig. 11(a) for the uniform distribution and Fig. 11(b) for the normal one. In GCSAC’s implementation, at iteration $k$, two random samples $(p_1, p_2)$ are chosen for estimating a cylinder as MLESAC. $p_1$ is kept and found a sample $p_2$ that 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 nearly that calculated from the ground-truth model. Because the inlier is high, the iteration $k$ is reduced as Eq. 7. GCSAC therefore is faster convergence. Fig. 13 shows final results of the estimated with 50% inlier synthesized dataset. Consequently, effectiveness of the "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 and consist of many noises and are challenging with different sizes. The first dataset [22] contains calibrated RGB-D data collected by a MS Kinect Version 1 of 111 indoor scenes. To adapt with 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., scene 2th, 4th, 9th, 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. It is noticed that ground-truths of the cylindrical objects in these datasets are manually prepared using a visualization tool of PCL library. Fig. 14(c) shows an ground-truth preparation example that a cylinder is specified by a line connecting two selected points on the top of the interested object. The third dataset is collected by ourself in indoor environments (e.g., cafeteria, sharing room) 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 are in a range from 3.5cm to 4.5cm with various heights (from 6.0 cm to 20 cm). A MS. Kinect Version 1 is mounted on the chest of a person who moves around a table. The dataset consists of 8 scenarios in which each scenario includes about 200 frames. In addition, we put some contaminated objects such as boxes (10.0cm × 30.0cm) besides the cylindrical objects. This dataset is built to adapt the context of practical application that supports visually impaired people finding an interested object.
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 are remained 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 $\gamma$) 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 measuring quantitative indexes, we used three following measurements:

- Let denote $E_a$ (degree) the different angle between the main direction of the estimated cylinder $\gamma_e$ and the normal vector of table plane $z_t$.

$$E_a = |\gamma_e - z_t|$$  \hspace{1cm} (10)

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

$$E_r = \frac{|R_e - R_g|}{R_g} \times 100\%$$  \hspace{1cm} (11)

- The processing time $t_p$ is 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 roles of the context’s constraints (as described in Section 3.5), the quantitative indexes are measured without using the proposed constraints and with using the constraints. In experiments, we fixed thresholds of the estimators with $T = 0.01$ (or 1cm), $w_t = 0.1$, $A_t = 20$ degrees. $T$ is a distance threshold to set a data point to be an inlier or outlier. For fair evaluations, $T$ 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 the 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
Fig. 17. (a)-(b) Some examples of the fitting results from the second, third dataset. In these scenes, there are more than one cylinder objects. They are marked in red, green, blue and yellow, so on. The estimated cylinders include radius, position (a center of the cylinder), main axis direction. The height can be computed using a 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.

<table>
<thead>
<tr>
<th>Dataset/Method</th>
<th>without the context’s constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_a$(deg.)</td>
</tr>
<tr>
<td>First dataset</td>
<td>MLESAC</td>
</tr>
<tr>
<td></td>
<td>GCSAC</td>
</tr>
<tr>
<td>Second dataset</td>
<td>MLESAC</td>
</tr>
<tr>
<td></td>
<td>GCSAC</td>
</tr>
<tr>
<td>Third dataset</td>
<td>MLESAC</td>
</tr>
<tr>
<td></td>
<td>GCSAC</td>
</tr>
</tbody>
</table>

three datasets. Compared with MLESAC, the estimated objects fitted by GCSAC algorithm are more higher 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 ($E_a$ is from 45° to 47°) of the angle derivations as defined in Eq. 10, using the GCSAC, $E_a$ is more lower, from 10° for the first dataset to 2° for the third dataset. The computational time is clearly different from GCSAC and MLESAC. Comparing between 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, pink bottles, as illustrated in Fig.18.

The fact that $E_a$ and $E_r$ are still large errors 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 ground-truth one. It is noticed that the evaluation results reported in Table 1 come from the implementations in which GCSAC is deployed without using the context’s constraints to verify the estimated model. The effectiveness of the context’s constraints is shown in Fig. 19. Obviously, by using the context’s constraints, the estimated objects could be eliminated when a large angle error is observed. The deviation angle is reduced from about 40° to 12°. From this, the full model of objects is built more accurate for grasping. 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 green estimated cylindrical object has relative error of the estimated radius $E_r = 111.08\%$; (b) the blue estimated cylindrical object has relative error of the estimated radius $E_r = 165.92\%$.

Fig. 19. Angle errors $E_a$ of the fitting results using GCSAC with and without using the context’s constraint.
4.4 Discussion

Beyond proposing a new sampling strategy for a robust estimator, our final goal aims to develop the object-finding-aided system for visually impaired people. The entire procedures of the proposed framework consisting of collecting RGB and depth data from Kinect; table detection; fitting objects, requires 1.04s per frame. In these procedures, we do not down-sampling the data. Fig. 20 shows snap-shots from one minute video, taken from common scene in an indoor environment of the third dataset. The completed video and relevant scenes are available in the link http://mica.edu.vn/perso/Le-Van-Hung/videodemo/index.html. In complex scene, e.g., there are four cylindrical objects on a table. The proposed method successfully locates them in almost scenes. As consequence, the proposed method toward to 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 with 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, 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 constraints of the context 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 expand 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