# GCSAC: Geometrical Constraint SAmple Consensus for Primitive Shapes Estimation in 3-D Point Cloud

**Abstract:** Estimating parameters of a primitive shape from a 3-D point cloud data is a challenging problem due to data containing noises and computational time demand. In this paper, we present a new robust estimator (named GCSAC, Geometrical Constraint SAmple Consensus) aimed at solving such issues. The proposed algorithm takes into account geometrical constraints to construct qualified samples for the estimation. Instead of randomly drawing minimal subset of sample, explicit geometrical properties of the interested primitive shapes (e.g., cylinder, sphere and cone) are used to drive sampling procedures. At each iteration of GCSAC, the minimal subset sample is selected based on two criteria (1) It must ensure a consistency with the estimated model via a roughly inlier ratio evaluation; (2) The samples satisfy geometrical constraints of the interested objects. Based on the obtained good samples, model estimation and verification procedures of the robust estimator are deployed in GCSAC. Extensive experiments have been conducted on synthesized and real datasets for evaluation. Comparing with the common robust estimators of RANSAC family (RANSAC, PROSAC, MLESAC, MSAC, LO-RANSAC and NAPSAC), GCSAC outperforms in term of both the precision of the estimated model and computational time. The implementations of the proposed method and the datasets are made publicly available.

**Keywords:** Robust Estimator; Primitive Shape Estimation; RANSAC and RANSAC Variations; Quality of Samples; Point Cloud data.

## 1 Introduction

Estimating parameters of a primitive shape is a fundamental research in the fields of robotic and computer vision. The geometrical model of an interested object such as a plane, sphere, cylinder, cone, can be estimated by two to seven geometrical parameters. A Random Sample Consensus (RANSAC) (Fischler and Bolles, 1981) 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. For being more accurate, faster and more robust, the RANSAC family focuses on either a better hypothesis from random samples or higher accuracy of data satisfying the estimated model. In this paper, we propose to exploit geometrical constraints to obtain a qualified Minimal Sample Set (MSS), i.e. good samples. This sample set can be used to generate better hypotheses, and as a result estimated model is achievable.

Originally, a RANSAC paradigm draws randomly a MSS from a point cloud data without any assumptions. As result, RANSAC must run a relatively large number of iterations to find an optimal solution before stopping criterion. To improve performances, RANSAC-based methods (Choi et al., 2009) focus on either a better hypothesis from random samples or higher quality of the samples satisfying the estimated model. In this paper, we tackle a new sampling procedure which utilizes geometrical constraints to qualify a MSS. We examine the proposed method with common primitive shapes such as a cylinder, a sphere, and a cone.

In the proposed robust estimator, named GCSAC (Geometrical Constraint SAmple Consensus), a MSS consists of samples which ensure two criteria: (1) the samples must be consistent with the estimated model based an approximate inlier ratio evaluation; (2) the samples must satisfy geometrical constraints of the interested objects (e.g., cylinder, sphere, cone constraints). More specifically, the key idea of the proposed method is guiding minimal sample set based on searching normal vector constraints of the geometric models. The constraints are derived from the explicit geometrical properties of the interested shapes. The good samples of a certain MSS are highly expected to generate a consensus set. Consequently, the number of iterations could be adaptively updated (similar to the termination manner of the adaptive RANSAC (Hartley and Zisserman, 2004)). In GCSAC, we utilize the maximum log-likelihood of MLESAC algorithm (Torr and Zisserman, 2000) to evaluate the estimated model. Finally, the effectiveness of the proposed method is confirmed by fitting common shapes such as a sphere, cylinder and cone in both synthesized and public datasets. In these evaluations, GSSAC performances are compared with common RANSACs as original RANSAC (Fischler and Bolles, 1981), PROSAC (Chum and Matas, 2005), MLESAC (Torr and Zisserman, 2000), LO-SAC (Chum et al., 2003), NAPSAC (Myatt et al., 2002). The implementations of the proposed method and the datasets are made publicly available (Authors, 2017)

## 2 RELATED WORK

For a general introduction and performances of RANSAC family, readers can refer to good surveys in (Raguram et al., 2008; Choi et al., 2009). 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 the primitive shapes.

For the first category, 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 (Chum and Matas, 2005) 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 (Chum et al., 2003) and its fixed version LO$^+$-RANSAC (Lebeda et al., 2012) add local optimization steps within RANSAC to improve accuracy. To speed up the computation, adaptive RANSAC (Hartley and Zisserman, 2004) 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 (Hartley and Zisserman, 2004) should be explored. Recently, USAC (Raguram et al., 2013) introduces a new frame-work for a robust estimator. In the USAC frame-work, 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 (Kohei et al., 2016) proposes to use geometric verification within a RANSAC frame-work. 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. The RANSAC-based algorithm used in the method of Chen et al. (Chen et al., 1999) and Aiger et al. (Aiger et al., 2008) for registering of partially overlapping range images and partial surfaces of a 3D object.

For primitive shape estimation from 3-D point clouds, readers can refer to a survey on feature-based techniques (Alhamzi and Elmogy, 2014). Relevant fitting techniques, for instance, multiscale super-quadric fitting in (Duncan et al., 2013), Hough transform in (Osselman et al., 2004), are commonly used. Marco et al. (Marco et al., 2014), Anas et al. (Anas et al., 2013) used the 3-D Hough Transform to estimate, extract sphere from point cloud data. However, the robust estimators (e.g., RANSAC family (Choi et al., 2009)) are always preferred techniques. Original RANSAC (Fischler and Bolles, 1981) demonstrates itself robust performances in estimating cylinders from range data. In (Trung-Thien et al., 2015), normal vectors and curvature information are used for parameter estimation and extraction of cylinders. The cylindrical objects are also interested in the analytic geometrical techniques. The authors in (Garcia, 2009) and (Schnabel et al., 2007) formulate primitive shapes (e.g., line, plane, cylinder, sphere, cone) using two to seven parameters such as a cylinder has seven parameters, a sphere has four parameters, a cone has seven parameters, etc. Schnabel et al. (Schnabel et al., 2007) defines primitive shapes through some samples and their normal vectors. In this study, geometrical analysis of a cylinder in (Schnabel et al., 2007) 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 Overview of the proposed robust estimator (GCSAC)*

To estimate parameters of a primitive shape, there are two main steps *hypothesis-and-verification* in the RANSAC-based algorithms (Choi et al., 2009) (e.g., RANSAC, MLESAC, PROSAC, MSAC, LOSAC, NAPSAC, .etc). First, to estimate the model, either drawing randomly a Minimal Sample Set (MSS)(RANSAC, MLESAC, MSAC) or semi-random (PROSAC) or using constraints of the sample's distribution (NAPSAC) is performed; Then, the estimated model is validated via a certain criteria. This scheme is repeated $K$ iterations to choose the best model. These procedures are shown in the top panel of Fig. 1. In this study, the proposed robust estimator (GCSAC) will be deployed for several types of the primitive shapes such as cylinder, sphere, cone. The implementations of GCSAC algorithm are shown in the bottom panel of Fig. 1. At the initial iterations, the proposed GCSAC constructs a MSS by the random sampling scheme and using a low inlier threshold to validate the estimated model. After only (few) random sampling iterations, the candidates of good samples could be initialized due to a weak-requirement of the inlier ratio. Once initial MSS is established, its samples will be updated by searching the qualified ones (or good samples) so that the geometrical constraints of the interested model is satisfied. The estimated model is evaluated according to Maximum Log-Likelihood criteria as MLESAC (Torr and Zisserman, 2000). The final step is to determine the termination condition, which is adopted from the adaptive RANSAC algorithm (Hartley and Zisserman, 2004). Once the

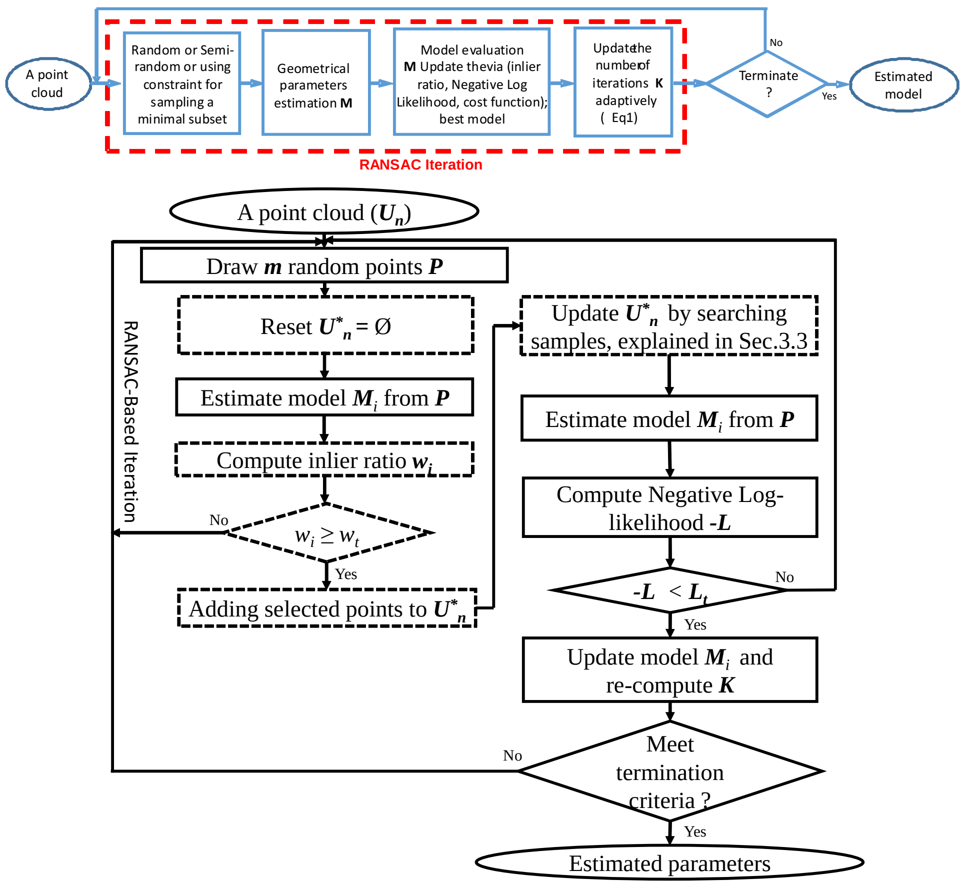


**Figure 1** Top panel: Over view of RANSAC-based algorithm. Bottom panel: A diagram of the GCSAC's implementations.

higher inlier ratio is obtained, the criterion termination $K$ for determining the number of sample selection is updated by:

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

where $p$ is the probability to find a model describing the data, $m$ is the minimal number of samples to estimate a model, $w$ is percentage of inliers in the point cloud. While $p$ is usually set by 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.

Obviously, defining the geometrical criteria, which are to search the good samples, is the most important. Let denote $U_n^*$ storing $m$ sample points to estimate a model, where $m = 2$ for cylinders or spheres, and $m = 3$ for cones. Based on the idea of the adaptive RANSAC (Hartley and Zisserman, 2004) to probe initial samples, GCSAC starts from roughly select of initial good samples. To initialize a set $U_n^*$, we assume that the worst case of inlier ratio (or a weak-requirement of the inlier ratio, $w_t = 0.1$ or $10\%$ inlier) is pre-determined. As expectedly, a consensus set containing at least $10\%$ inlier is easily found. Once a MSS is found, $m-1$ samples is randomly selected for preserving and a remaining one, $m^{th}$, will be replaced by better one so that the set $U_n^*$ is the best satisfied *geometrical constraints* of the interested shape. Consequently, the model is estimated from *good samples* that directly effects to estimating the inlier ratio at current iteration. At next iteration, $U_n^*$ is reset ($U_n^* = \varnothing$) for other estimations. If $U_n^*$ could not be initialized (or there is none of iterations which the condition $w_i \geq w_t$ is not satisfied), GCSAC algorithm degrades to the original RANSAC. The geometrical principles and constraints of a primitive shape are explained in Section 3.2.

## *3.2 Geometrical analyses and constraints for qualifying good samples*

In following sections, the principles of 3-D the primitive shapes are explained. Based on this geometrical analysis, the related constraints are given to select good samples.

### *3.2.1 Geometrical analysis for cylindrical objects*

For geometrical analysis of a cylinder object, we adopted the analysis given by (Schnabel et al., 2007). 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 vector $\gamma_c$ of main axis direction; and its radius $R_c$. The geometrical relationships of the cylindrical parameters are shown in Fig. 2.

A cylinder is estimated from two points $(p_1, p_2)$ (two grey-squared points in Fig. 2(a)) and their corresponding normal vectors $(\mathbf{n}_1, \mathbf{n}_2)$ (blue lines in Fig. 2(a)). The normal vector of any point is computed following the approach in Holz et al. (Dirk Holz et al., 2011). At each point $p_i$, k-nearest neighbors $k_n$ of $p_i$ are determined within a radius $r$. The normal vector of $p_i$ is therefore reduced to analysis of eigenvectors and eigenvalues of the covariance matrix $C$, that is given by:

$$C = \sum_{i}^{k_n} (p_i - p_{av})(p_i - p_{av})^T \qquad CV_j = \lambda_j V_j, \qquad j \in 0, 1, 2 \tag{2}$$

where $p_{av} = \frac{1}{k_n}\sum_i^{k_n} p_i$ represents a 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. (2). The first eigenvector $V_0$ corresponding to least eigenvalue $\lambda_0$ will be the normal vector at sample point $p_i$.

Let $\gamma_c$ is the main axis direction (pink line in Fig. 2(a)) of the cylinder. It is estimated by $\gamma_c = \mathbf{n}_1 \times \mathbf{n}_2$. To estimate the center point $I_c$, we project two parametric lines $L1 = p_1 + t\mathbf{n}_1$ and $L2 = p_2 + t\mathbf{n}_2$ along the axis onto the $PlaneY$ plane (see a green plane in Fig. 2(b)). The normal vector of this plane is estimated by a cross product of $\gamma_c$ and $\mathbf{n}_2$ vectors ($\gamma_c \times \mathbf{n}_2$). The centroid point $I_c$ (see a red point in Fig. 2(d)) is the intersection of $L1, L2$ (see two green line in Fig. 2(c)). The radius $R_c$ is set by the distance between $I_c$ and $p_1$ on that plane. The estimated cylinder from a point cloud is illustrated in Fig. 2(d). The height of the estimated cylinder is normalized to 1. It is noticed that the estimated cylinder in Fig. 2(d) is a wrong estimation because the hypothesis in this case consists of an inlier $p_2$ and an outlier $p_1$ sample.

**The geometrical constraints for cylindrical objects**: To deploy the geometrical constraints for cylindrical objects, let's examine the following example. In each hypothesis,

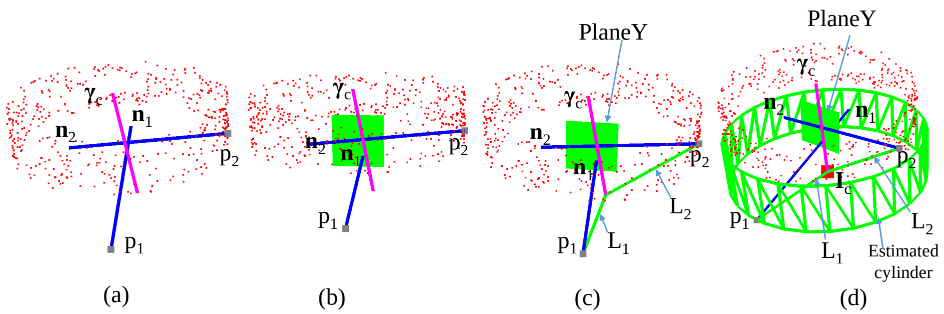


**Figure 2** Geometrical parameters of a cylindrical object. Red points are defined inlier points in the generated dataset, which has 15% inlier point. (a)-(c) Explanation of the geometrical analysis to estimate a cylindrical object. (d) The result of the estimated cylinder from a point cloud (green estimated cylinder). In this figure, the selected point $p_1$ is an outlier point, it makes, the centroid point of estimated cylinder deviates.

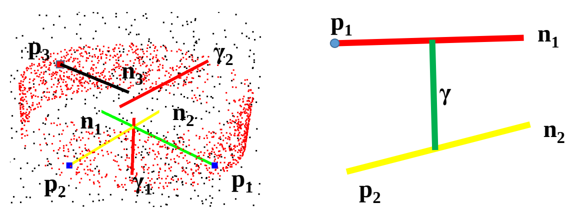


**Figure 3** Illustration of the geometrical constraints applied in GCSAC.

a MSS could be two within three samples $p_1$, $p_2$, and $p_3$, as shown in Fig. 3(a). In the 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. 2(a), the direction of the axis $\gamma_2$ 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 criteria (or search good samples) aims to update the initial samples (expectedly, $p_3$ should be replaced 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. 3(b). In the other words, $\mathbf{n}_1$ needs to be perpendicular to $\mathbf{n}_2^*$ where $\mathbf{n}_2^*$ is a projection of $\mathbf{n}_2$ onto a plane $\pi$ whose normal vector $n_\pi$ is $\mathbf{n}_{PlaneY} \times \mathbf{n}_1$. This observation leads to the criteria below:

$$c_p = \underset{p_2 \in \{U_n \backslash p_1\}}{\operatorname{argmin}} \{\mathbf{n}_1 \cdot \mathbf{n}_2^*\} \tag{3}$$

If $c_p$ is close to 0 then $\mathbf{n}_1$ and $\mathbf{n}_2^*$ are orthogonal. It is noticed that in the example as shown in Fig. 3(a), the projection of $\mathbf{n}_3$ onto a plane $\pi$ should be parallel to $\mathbf{n}_1$. Therefore the dot product $\mathbf{n}_1 \cdot \mathbf{n}_3^*$ is a large scalar value.

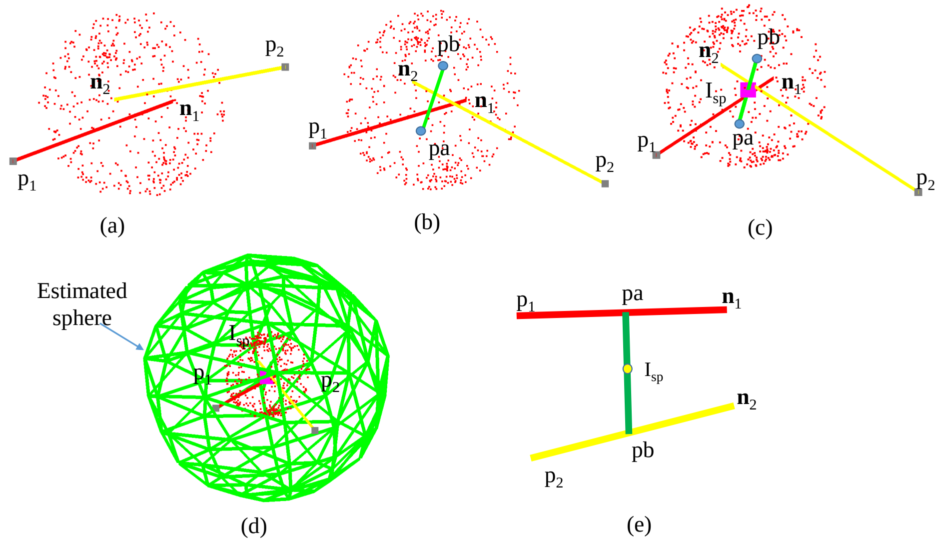


**Figure 4** Estimating parameters of a sphere from 3-D points. Red points are inlier points. In this figure, $p_1, p_2$ is two selected samples for estimating a sphere (two gray points), they are outlier points. Therefore, the estimated sphere is wrong of centroid and radius (see green sphere (d)).

### *3.2.2 Geometrical analysis for a spherical object*

A sphere is determined by the following parameters: a centroid point which is denoted as $I_{sp}(x_0, y_0, z_0)$; its radius $R_{sp}$. To estimate sphere's parameters, Schnabel et al. (Schnabel et al., 2007) propose to use two points $(p_1, p_2)$ with their corresponding normal vectors $(\mathbf{n}_1, \mathbf{n}_2)$ (see Fig. 4(a)). The centroid $I_{sp}$ (a pink point Fig. 4(c)) is a middle point of the shortest line (a green line of Fig. 4(b)) which segments two lines given by $(p_1, \mathbf{n}_1)$ and $(p_2, \mathbf{n}_2)$. This segmented line is illustrated by $\overline{papb}$ in Fig. 4(b). The radius $R_{sp}$ is determined by averaging the distance of $I_{sp}$ to $p_1$ and $I_{sp}$ to $p_2$. Illustration of the estimated sphere is shown in Fig. 4(d).

**The geometrical constraints for fitting spherical objects**: As above denoted, a sphere is estimated from two points $(p_1, p_2)$ and their normal vectors $(\mathbf{n}_1, \mathbf{n}_2)$. In GCSAC, once set $U_n^*$ consisting of the initial good samples is conducted, we store $p_1$ and search $p_2$ in the whole point cloud. We observe that to generate a sphere, the triangle $(p_1 I_{sp} p_2)$ should be isosceles, as shown in Fig. 4(e). Consequently, this observation forms a geometrical constraint for searching $p_2$ as following:

$$sh_p = \underset{p_2 \in \{U_n \backslash p_1\}}{\operatorname{argmin}} \{(|p_1 I_{sp}| - |p_2 I_{sp}|)\} \tag{4}$$

The geometrical constraints in Eq. (4) means that if $sh_p$ is close to 0 then the triangle $p_1 I_{sp} p_2$ is nearly isosceles one.

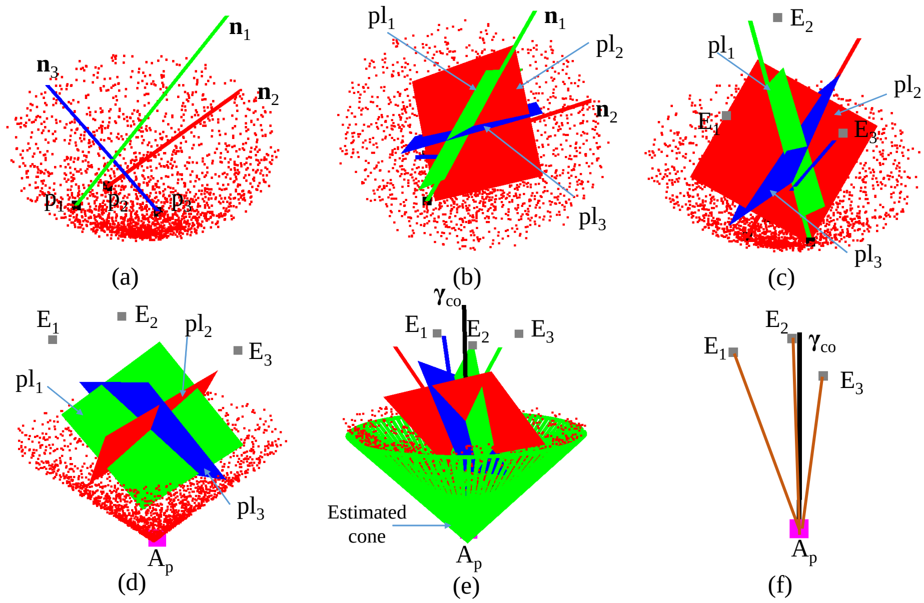

**Figure 5** (a) Estimating parameters of a cone from 3-D points by (Schnabel et al., 2007); (b) Illustrating constraint of estimating the good cone.

### 3.2.3 Geometrical analysis for a conical object

A cone is determined by the following parameters: an apex on the cone axis which is denoted as $A_p(x_0, y_0, z_0)$; a vector of the main direction axis denoted as $\gamma_{co}$; an opening angle of the cone denoted as $\vartheta$. To estimate these parameters, Schnabel et al. (Schnabel et al., 2007) utilize three points $(p_1, p_2, p_3)$ and their normal vectors $(\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3)$ (see Fig. 5(a)). Especially, to identify the position of the apex $A_p$ (see a pink point of Fig. 5(d)), intersections of the three planes which are defined by points and normal vector pairs (red $(pl_1)$, green $(pl_2)$, blue $(pl_3)$ planes in Fig. 5(b)), are defined by Eq. 5.

$$A_p = Intersection(pl_1, pl_2, pl_3)/ \\ pl_1(p_1, \times(\mathbf{n}_2, \mathbf{n}_3)); pl_2(p_2, \times(\mathbf{n}_1, \mathbf{n}_3)); pl_3(p_3, \times(\mathbf{n}_1, \mathbf{n}_2)); \quad (5)$$

Definitions of three points $E_1, E_2, E_3$ (see three grey points in Fig. 5(c)) are given by Eq. (6):

$$E_1 = A_p + \frac{p_1 - A_p}{|| \, p_1 - A_p \, ||}; E_2 = A_p + \frac{p_2 - A_p}{|| \, p_2 - A_p \, ||}; E_3 = A_p + \frac{p_3 - A_p}{|| \, p_3 - A_p \, ||} \quad (6)$$

The normal vector of a plane is defined by three points $E_1, E_2, E_3$, it is the direction of main axis $\gamma_{co}$ which is marked as a black line of Fig. 5(e). The opening angle $\vartheta$ is identified by Eq. 7. The estimated cone is shown in Fig. 5(e).

$$\vartheta = \frac{\sum_i arcos(p_i - A_p)\gamma_{co}}{3}; (i = 1, 2, 3) \quad (7)$$

**Table 1** The characteristics of the synthesized datasets for cylinder, sphere, cone objects

| Dataset | Characteristics of the generalized data | | | |
|---|---|---|---|---|
| | Height/ Radius | Direction of main axis | Spatial distribution of inliers | Spread of outliers |
| dC1, dC4 | 1 /2 | parallel with the z-axis | Around of a cylinder | [-3, 3], [-4, 4] |
| dSP1, dSP4 | /1 | | Around of a sphere | [-3, 3], [-4, 4] |
| dCO1, dCO4 | 1/1 | parallel with the z-axis | Around of a cone | [-3, 3], [-4, 4] |
| dC2, dC5 | 1 /2 | parallel withthe y-axis | Around of a cylinder | [-3, 3], [-4, 4] |
| dSP2, dSP5 | /1 | | Around of a sphere | [-3, 3], [-4, 4] |
| dCO2, dCO5 | 1/1 | parallel with the y-axis | Around of a cone | [-3, 3], [-4, 4] |
| dC3, dC6 | 1 /2 | parallel with the y-axis | one half of a cylinder | [-3, 3], [-4, 4] |
| dSP3, dSP6 | /1 | | one half of a sphere | [-3, 3], [-4, 4] |
| dCO3, dCO6 | 1 /1 | parallel with the y-axis | one half of a cone | [-3, 3], [-4, 4] |

**The geometrical constraints for fitting conical objects**: Similar to the cylinder and sphere, once $U_n^*$ consisting of the initial good samples, we store $p_1, p_2$ and search a remaining sample $p_3$. The $p_3$ will be replaced by a new sample which ensures so that the tetrahedra ($A_pE_1E_2E_3$) has all surfaces being near identical. It searches in $U_n$ and is difference $p_1$, $p_2$. In other words, its surfaces should be isosceles triangles, defined by:

$$co_p = \underset{p_3 \in \{U_n \setminus \{p_1, p_2\}\}}{\text{argmin}} \{(|| A_pE_1 || - || A_pE_2 ||) - (|| A_pE_1 || - || A_pE_3 ||)\} \quad (8)$$

If $co_p$ is close to 0 then the triangles $(A_pE_1E_2)$ and $(A_pE_1E_3)$ and $(A_pE_2E_3)$ are isosceles triangles at a apex $A_p$ and are identical, as shown in Fig. 5(f).

## 4 EXPERIMENTAL RESULT

### *4.1 Evaluation Datasets*

We evaluate performances of GCSAC on two types of datasets. The first is synthesized datasets and second is realistic ones. These datasets consists of cylinders, spheres and cones. For each interested object, the synthesized dataset consists of six different subsets. Characteristics of each subset are described in Table 1. Major differences could be the main axis's orientation, $\sigma$ of the normal distribution for generating outlier/inlier data; or the spatial distribution of inliers.

For the cylinder dataset *('first cylinder')*, they are denoted from $dC_1$ to $dC_6$. In each subset $dC_i$, inlier ratio is increased by a step of 5% in a range from 15% to 80 %. Therefore, there are fourteen point clouds. They are denoted $dS_1$ to $dS_{14}$. A point cloud $dS_i$ consists of 3000 sample points. To generate cylinder dataset, an inlier data point $(x_i, y_i, z_i)$ of $dS_i$ is lying on a cylinder surface which is generated as follow: $x_i = cos(\theta_i), z_i = sin(\theta_i), y_i$ is randomly selected in [0, 1], $\theta_i$ is randomly selected from [0, $2\pi$]. Outliers are generated randomly in a range as given in the last column in Table 1. Fig. 6(a) illustrates the synthesized data of $dC_1, dC_2, dC_3$ whose inlier ratio equals 50%.

Similar to cylinder dataset, point clouds of the sphere dataset *'first sphere'* are denoted from $dSP_1$ to $dSP_6$. These point clouds are generated from surface of a true sphere: $x^2 + y^2 + z^2 = 1$. Some illustrations of the synthesised sphere are presented in Fig. 6. Point

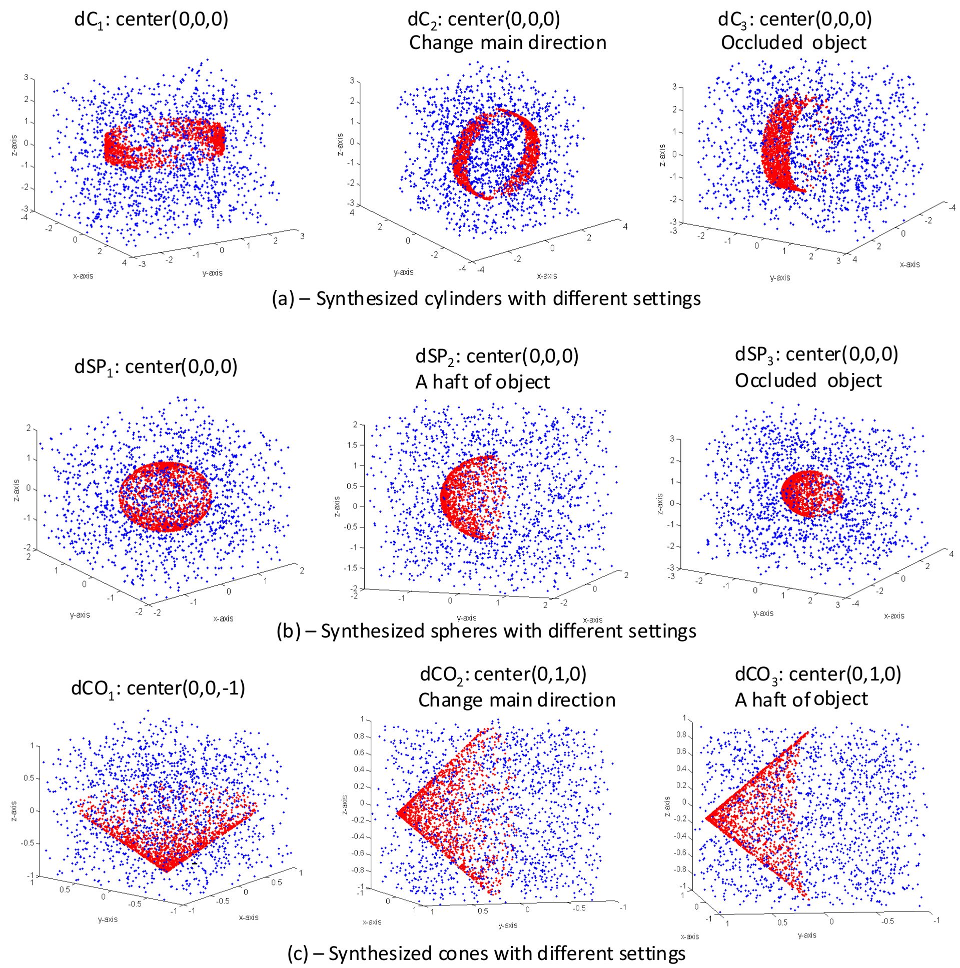


**Figure 6** Illustrations of three synthesized datasets with 50% inlier ratio. (a) $dC_1, dC_2, dC_3$ point clouds of the cylinders, (b) $dSP_1, dSP_2, dSP_3$ point clouds of the spheres (c) $dCO_1, dCO_2, dCO_3$ point clouds of the spheres The red points are inliers, whereas blue points are outliers.

clouds of the cone dataset *'first cone'* are denoted from $dCO_1$ to $dCO_6$. We also generate random points of the cone as outliers. They are illustrated in Fig. 6(c).

In addition, we evaluate the proposed method on real datasets. For the cylindrical objects, the dataset is collected from a public dataset (Lai et al., 2011) which contains 300 objects belonging to 51 categories. It named *'second cylinder'*. In this study, we collect only videos consisting of the cylindrical objects. Totally, the cylinder dataset consists of 8 coffee mugs, 14 food cans, 5 food cups, 6 soda cans. Fig. 8 shows some instances of the collected cylinders in the second dataset. For the spherical object, the dataset consists of two balls collected from four real scenes. Each scene has been included 500 frames. It named *'second sphere'*. This is a public sphere dataset collected in (Le et al., 2016). The point clouds of balls, as illustrated in Fig. 7(a), are manually separated from other objects in a scene such as table


**Figure 7** (a) Illustrating the separating the point cloud data of a ball in the scene. (b) Illustrating the point cloud data of a cone and preparing the ground truth of evaluating the fitting a cone.


**Figure 8** Examples of four cylindrical-like objects collected from the *'second cylinder'* dataset.

plane, wall, floor, etc. Finally, point cloud data of the cone objects, named *'second cone'*, is collected from dataset given in (Scharstein and Szeliski, 2003). To prepare ground-truth data, we used the stereo data to segment each cone object and convert them to point cloud data, as illustrated in Fig. 7(b).

### *4.2 Evaluation Measurements*

Let denote a general form for different interested objects (e.g., cylinder, sphere, cone) as following. The ground-truth of the interested object is $\mathbf{M}_t(x_t, y_t, z_t, r_t, a_t, o_t)$ and the estimated one is $\mathbf{M}_e(x_e, y_e, z_e, r_e, a_e, o_e)$ where $(x_t, y_t, z_t)$, $(x_e, y_e, z_e)$ are the coordinates of the center points of a cylinder, sphere, or cone; More specially, $r_t, r_e$ are the radius of a cylinder and sphere, respectively. $o_t, o_e$ are opening angles of the estimated cone and ground-truth one, respectively. Parameters $a_t, a_e$ are the angles between the main axis of the estimated shapes and the ground-truth ones. To evaluate the performance of the proposed method, we use following measurements:

- Let denote the relative error $E_w$ of the estimated inlier ratio. The smaller $E_w$ is, the better the algorithm is.

$$E_w = \frac{|w - w_{gt}|}{w_{gt}} \times 100 \tag{9}$$

where $w_{gt}$ is the defined inlier ratio of ground-truth; $w$ is the inlier ratio of the estimated model.

$$w = \frac{\text{number of inliers}}{\text{number of samples}} \tag{10}$$

- The total distance errors $S_d$ (Faber and Fisher, 2001) is calculated by summation of distances from any point $p_j$ to the estimated model $M_e$. $S_d$ is defined by:

$$S_d = \sum_{j=1}^{N} d(p_j, M_e) \tag{11}$$

- The processing time $t_p$ is measured in milliseconds (ms). The smaller $t_p$ is the faster the algorithm is.
- The relative error of the estimated center (only for the synthesized datasets) $E_d$: is an Euclidean distance between the estimated center $E_e$ and a ground-truth one $E_{gt}$. $E_d$ is defined by:

$$E_d = |E_e - E_{gt}| \tag{12}$$

- The relative error of the estimated radius (for evaluating cylinders and spheres) $E_r$: is the difference between the estimated radius $r_e$ and the ground-truth one $r_{gt}$. $E_r$ is defined by:

$$E_r = \frac{|r_e - r_{gt}|}{r_{gt}} \times 100\% \tag{13}$$

  For the conical object, $E_r$ means the opening angle error of the estimated cone. It is defined by:

$$E_r = \frac{|o_e - o_{gt}|}{o_{gt}} \times 100\% \tag{14}$$

- Let denote $E_a$ infer a difference between the estimated angle $a_e$ and ground-truth one $a_{gt}$. $E_a$ (Kwon et al., 2003) is calculated by:

$$E_a = |a_e - a_{gt}| \tag{15}$$

The proposed method (GCSAC) is compared with six common ones in RANSAC family. They are original RANSAC (Fischler and Bolles, 1981), PROSAC (Chum and Matas, 2005), MLESAC (Torr and Zisserman, 2000), MSAC (Torr and Murray, 1997), NAPSAC (Myatt et al., 2002), LO-RANSAC (Chum et al., 2003). For setting the parameters, we fixed thresholds of the estimators with $T = 0.05$ (or 5cm), $w_t = 0.1$, $s_r = 3$ (cm). $T$ is a distance threshold to set a data point to be inlier or outlier. $s_r$ is the radius of a sphere when using NAPSAC algorithm. For the fair evaluations, $T$ is set equally for all seven fitting methods.

The proposed method 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.

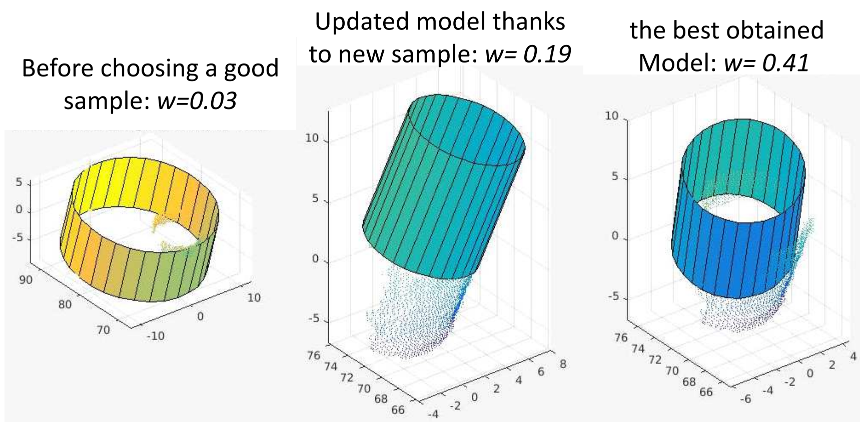


**Figure 9** An illustration of GCSAC's at a $k^{th}$ iteration to estimate a coffee mug in the second dataset. Left: the fitting result with a random MSS. Middle: the fitting result where the random samples are updated due to applying the geometrical constraints. Right: the current best model.

## *4.3 The evaluation results*

The performances of each method on the synthesized datasets are reported in Table 2. As reported, GCSAC obtains the highest accuracy and lowest computational time for whole three synthesized datasets. As shown by $E_w$ indexes of three types of primitive shapes, even using same criteria as MLESAC, the proposed GCSAC obtains better estimated model comparing with original MLESAC algorithm . Although $E_w$ of the sphere dataset is still high ($E_w = 19.44\%$), this result is still better than the result of the compared methods. Among the RANSAC-based variations, it is interesting that original RANSAC gives stable results for three interested shapes. However, original RANSAC requires a high computational time. The proposed GCSAC estimates the models slightly better than the original RANSAC, but it is lower in term of the computational time. To debug how GCSAC work, Fig. 9 illustrates affects of the updating a good sample in order to estimate a cylinder. By randomly drawing MSS samples, a RANSAC-based algorithm can generate a failed candidate, as shown in Fig. 9(a). However, once these samples are updated by the searching *good sample* procedure, a better model could be estimated (inlier rate of 0.19 refer to the current best model in Fig. 9(c)).

To intuitively visualize the performances of GSSAC and other six RANSAC-based algorithms, Fig.10, Fig.11, and Fig.12 show the fitting results in synthesised datasets of cylindrical, spherical, cone objects, respectively, which consist of only 15% inlier data. These illustrations therefore confirm the proposed geometrical constraints are working well with different primitive shapes.

For evaluating real datasets, the experimental results are reported in Table 3 for the cylindrical objects. Table 4 reports fitting results for spherical and cone datasets. It is noticed that these datasets consist of natural scenes which are taken from different viewpoints and various types/sizes of the interested objects. For the results of cylindrical objects, all of the evaluations show that GCSAC outperforms the MLESAC method. Especially, the estimated inlier ratio ($w$), and total distance error ($S_d$) confirm that fitting results are fairly good with

**Table 2** The average evaluation results of synthesized datasets. The synthesized datasets were repeated 50 times for statistically representative results.

| Dataset/ Method | Measure | **RANSAC** (Fischler and Bolles, 1981) | **PROSAC** (Chum and Matas, 2005) | **MLESAC** (Torr and Zisserman, 2000) | **MSAC** (Torr and Murray, 1997) | **LOSAC** (Chum et al., 2003) | **NAPSAC** (Myatt et al., 2002) | **GCSAC** |
|---|---|---|---|---|---|---|---|---|
| *'first cylinder'* | $E_w$ (%) | 23.59 | 28.62 | 43.13 | 10.92 | 9.95 | 61.27 | **8.49** |
| | $S_d$ | 1528.71 | 1562.42 | 1568.81 | 1527.93 | 1536.47 | 3168.17 | **1495.33** |
| | $t_p(ms)$ | 89.54 | 52.71 | 70.94 | 90.84 | 536.84 | 52.03 | **41.35** |
| | $E_d(cm)$ | 0.05 | 0.06 | 0.17 | 0.04 | 0.05 | 0.93 | **0.03** |
| | $E_A(deg.)$ | 3.12 | 4.02 | 5.87 | 2.81 | 2.84 | 7.02 | **2.24** |
| | $E_r(\%)$ | 1.54 | 2.33 | 7.54 | 1.02 | 2.40 | 112.06 | **0.69** |
| *'first sphere'* | $E_w(\%)$ | 23.01 | 31.53 | 85.65 | 33.43 | 23.63 | 57.76 | **19.44** |
| | $S_d$ | 3801.95 | 3803.62 | 3774.77 | 3804.27 | 3558.06 | 3904.22 | **3452.88** |
| | $t_p(ms)$ | 10.68 | 23.45 | 1728.21 | 9.46 | 31.57 | **2.96** | 6.48 |
| | $E_d(cm)$ | **0.05** | 0.07 | 1.71 | 0.08 | 0.21 | 0.97 | **0.05** |
| | $E_r(\%)$ | 2.92 | 4.12 | 203.60 | 5.15 | 17.52 | 63.60 | **2.61** |
| *'first cone'* | $E_w(\%)$ | 24.89 | 37.86 | 68.32 | 40.74 | 30.11 | 86.15 | **24.40** |
| | $S_d$ | 2361.79 | 2523.68 | 2383.01 | 2388.64 | 2298.03 | 13730.53 | **2223.14** |
| | $t_p(ms)$ | 495.26 | 242.26 | 52525 | 227.57 | 1258.07 | 206.17 | **188.4** |
| | $E_A(deg.)$ | 6.48 | 15.64 | 11.67 | 15.64 | 6.79 | 14.54 | **4.77** |
| | $E\_r(\%)$ | 20.47 | 17.65 | 429.44 | 17.31 | 20.22 | 54.44 | **17.21** |

**Table 3** Experimental results on the *'second cylinder'* dataset. The experiments were repeated 20 times, then errors are averaged

| **Dataset/ Measure** | **Method** | $w$ (%) | $S_d$ | $t_p$ $(ms)$ | $E_r$ (%) |
|---|---|---|---|---|---|
| *'second cylinder'* (coffee mug) | MLESAC | 9.94 | 3269.77 | 110.28 | 9.93 |
| | GCSAC | **13.83** | **2807.40** | **33.44** | **7.00** |
| *'second cylinder'* (food can) | MLESAC | 19.05 | 1231.16 | 479.74 | 19.58 |
| | GCSAC | **21.41** | **1015.38** | **119.46** | **13.48** |
| *'second cylinder'* (food cup) | MLESAC | 15.04 | 1211.91 | 101.61 | 21.89 |
| | **GCSAC** | **18.8** | **1035.19** | **14.43** | **17.87** |
| *'second cylinder'* (soda can) | MLESAC | 13.54 | 1238.96 | 620.62 | 29.63 |
| | GCSAC | **20.6** | **1004.27** | **16.25** | **27.7** |

GCSAC. Different from the evaluations on the synthesized datasets, $E_w$ is not available for the real datasets. The reason is that $w_g t$, a true inlier ratio, is not able to measure in the real scenes. As given by $w$ indexes in Table 4, the results of *'second sphere'* dataset are quite similar for all of the evaluation methods. We observe that the balls datasets have a small noise ratio. However, the $E_r$ of GCSAC is slightly better than others. For the *'second cone'* dataset, all of error indexes confirmed performances of the GCSAC versus others. Specially, the computation time $t_p$ of GCSAC is significantly lower than others' results. Figure 13(a)-(c) illustrate the fitting results of cylindrical, spherical and cone objects using GCSAC on the real datasets.

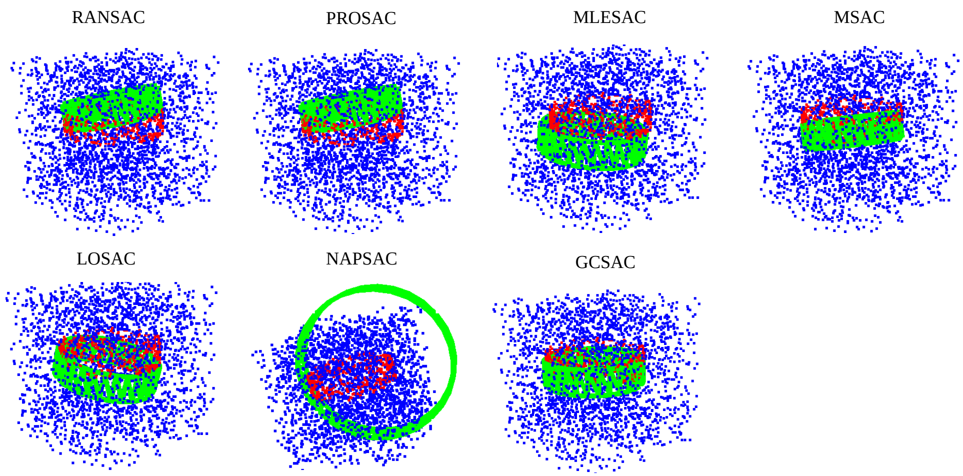


**Figure 10** Illustrations of the fitting results on the synthesized datasets of cylinder using GCSAC and other RANSAC variations. The datasets consists 45% inlier ratio. Red points are inliers, blue points are outlier, the estimated cylinder is marked by green points.

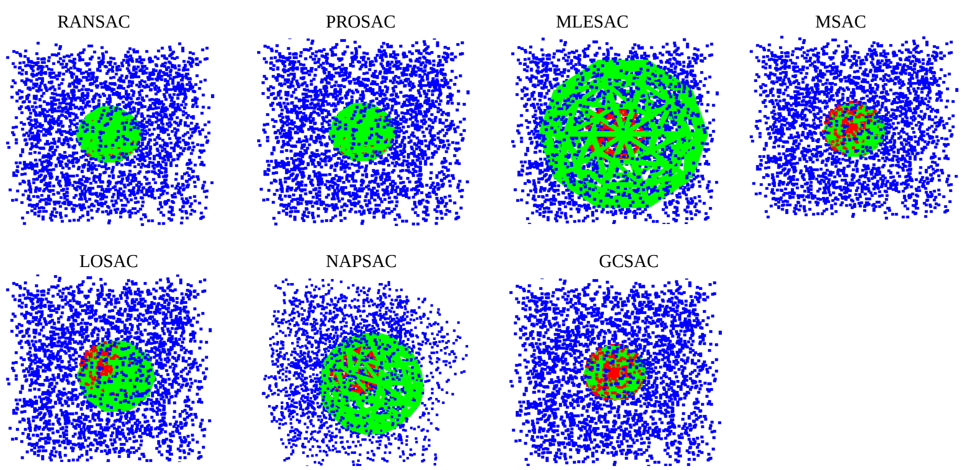


**Figure 11** Illustrations of the fitting results on the synthesized datasets of sphere using GCSAC and other RANSAC variations. The datasets consists 45% inlier ratio. Red points are inliers, blue points are outlier, the estimated sphere is marked by green points.

## 5 Conclusions

In this paper, we proposed GCSAC that is a new RANSAC-based robust estimator for fitting the primitive shapes from point clouds. The key idea of the proposed GCSAC was the combination of ensuring consistency with the estimated model via a roughly inlier ratio evaluation and geometrical constraints of the interested shapes. This strategy aimed to select good samples for the model estimation. The proposed method was examined with primitive shapes such as a cylinder, sphere and cone. The experimental datasets consisted

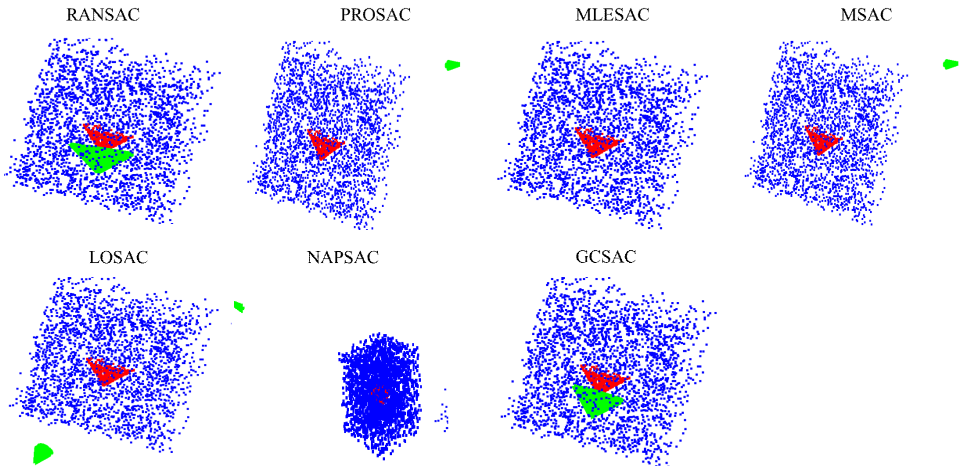


**Figure 12** Illustrations of the fitting results on the synthesized datasets of cone using GCSAC and other RANSAC variations. The datasets consists 45% inlier ratio. Red points are inliers, blue points are outlier, the estimated cone is marked by green points.

**Table 4** The average evaluation results on the *'second sphere'*, *'second cone'* datasets. The real datasets were repeated 20 times for statistically representative results.

| Dataset/ Method | Measure | **RANSAC** (Fischler and Bolles, 1981) | **PROSAC** (Chum and Matas, 2005) | **MLESAC** (Torr and Zisserman, 2000) | **MSAC** (Torr and Murray, 1997) | **LOSAC** (Chum et al., 2003) | **NAPSAC** (Myatt et al., 2002) | **GCSAC** |
|---|---|---|---|---|---|---|---|---|
| *'second sphere'* | $w(\%)$ | 99.77 | 99.98 | 99.83 | 99.80 | 99.78 | 98.20 | **100.00** |
| | $S_d$ | 29.60 | 26.62 | 29.38 | 29.37 | 28.77 | 35.55 | **11.31** |
| | $t_p(ms)$ | 3.44 | 3.43 | 4.17 | 2.97 | 7.82 | 4.11 | **2.93** |
| | $E_r(\%)$ | 30.56 | 26.55 | 30.36 | 30.38 | 31.05 | 33.72 | **14.08** |
| *'second cone'* | $w(\%)$ | 79.52 | 71.89 | 75.45 | 71.89 | 80.21 | 38.79 | **82.27** |
| | $S_d$ | 126.56 | 156.40 | 147.00 | 143.00 | 96.37 | 1043.34 | 116.09 |
| | $t_p(ms)$ | 10.94 | 7.42 | 13.05 | 9.65 | 96.37 | 25.39 | **7.14** |
| | $E_A(deg.)$ | 38.11 | 40.35 | 35.62 | 25.39 | 29.42 | 52.64 | **23.74** |
| | $E_r(\%)$ | 77.52 | 77.09 | 74.84 | 75.10 | 71.66 | 76.06 | **68.84** |

of synthesized, real datasets. The results of the GCSAC algorithm were compared to various RANSAC-based algorithms and they confirm that GCSAC worked well even the point-clouds with low inlier ratio. In the future, we continue to validate GCSAC on other geometrical structures and evaluate the proposed method with the real scenario for detecting multiple objects.

## References

Aiger, D., N. J. Mitra, and D. Cohen-Or
2008. 4-points congruent sets for robust surface registration. *ACM Transactions on Graphics*,

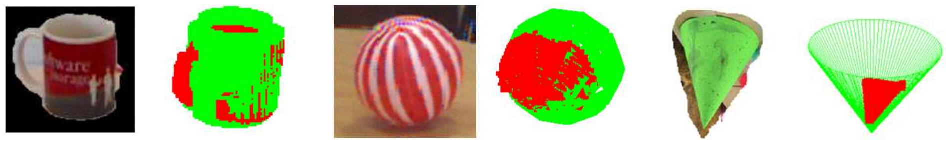


**Figure 13** Result fitting of some instances collected from the real datasets. (a) A coffee- mug; (b) A toy ball; (c) A cone object. In each sub-figure: left-panel is RGB image for a reference, right-panel is fitting result. Ground-truths are marked as red points; the estimated objects are marked as green points.

27(3):1–10.

Alhamzi, K. and M. Elmogy
2014. 3d object recognition based on image features : A survey. *International Journal of Computer and Information Technology*, 03(03):651–660.

Anas, A., N. Mark S, and C. John N
2013. Sphere detection in kinect point clouds via the 3d hough transform. In *International Conference on Computer Analysis of Images and Patterns*.

Authors
2017. `http://`. [ Due to double-blind review's policy of the Journal, URL of the external material is not given. It will be provided once the Editors/Reviewers request.].

Chen, C.-S., Y.-P. Hung, and J.-B. Cheng
1999. RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229 –1234.

Choi, S., T. Kim, and W. Yu
2009. Performance evaluation of ransac family. In *Procedings of the British Machine Vision Conference 2009*, Pp. 1–12. British Machine Vision Association.

Chum, O. and J. Matas
2005. Matching with prosac progressive sample consensus. In *Procedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Pp. 220–226.

Chum, O., J. Matas, and J. Kittler
2003. Locally optimized ransac. In *DAGM-Symposium*, volume 2781 of *Lecture Notes in Computer Science*, Pp. 236–243. Springer.

Dirk Holz, S., R. B. Rusu, and S. Behnke
2011. Real-Time Plane Segmentation Using RGB-D Cameras. In *LNCS (7416): RoboCup 2011 - Robot Soccer World Cup XV*, Pp. 306–317.

Duncan, K., S. Sarkar, R. Alqasemi, and R. Dubey
2013. Multiscale superquadric fitting for efficient shape and pose recovery of unknown objects. In *Procedings of the International Conference on Robotics and Automation (ICRA'2013)*.

Faber, P. and R. B. Fisher
2001. A Buyer's Guide to Euclidean Elliptical Cylindrical and Conical Surface Fitting. In *Procedings of the British Machine Vision Conference 2001*, Pp. 54.1–54.10.

Fischler, M. A. and R. Bolles
1981. Random sample consensus: A paradigm for model fitting with applications to image analysisand automated cartography. *Communications of the ACM*, 24(6):381–395.

Garcia, S.
2009. Fitting primitive shapes to point clouds for robotic grasping. *Master Thesis in Computer Science (30 ECTS credits) at the School of Electrical Engineering Royal Institute of Technology*, Pp. 1–100.

Hartley, R. I. and A. Zisserman
2004. *Multiple View Geometry in Computer Vision*, second edition. Cambridge University Press, ISBN: 0521540518.

Kohei, M., U. Yusuke, S. Shigeyuki, and S. Sato
2016. Geometric verification using semi-2d constraints for 3d object retrieval. In *Proceedings of the International Conference on Pattern Recognition (ICPR) 2012.*, Pp. 2339–2344.

Kwon, S.-W., K. A. Liapi, C. T. Haas, and F. Bosché
2003. Algorithms for fitting cylindrical objects to sparse range point clouds for rapid workspace modeling. In *Proceedings of the 20th ISARC*, Pp. 173–178.

Lai, K., L. Bo, X. Ren, and D. Fox
2011. A large-scale hierarchical multi-view RGB-D object dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, Pp. 1817–1824.

Le, V. H., T. L. Le, H. Vu, T. T. Nguyen, T. H. Tran, T. C. Dao, and H. Q. Nguyen
2016. Geometry-based 3-d object fitting and localization in grasping aid for visually impaired. In *The Sixth International Conference on Communications and Electronics*. IEEE-ICCE.

Lebeda, K., J. Matas, and O. Chum
2012. Fixing the locally optimized ransac. In *Proceedings of the British Machine Vision Conference 2012.*, Pp. 3–7.

Marco, C., V. Roberto, and C. Rita
2014. 3d hough transform for sphere recognition on point clouds. *Machine Vision and Applications*, P. 1877âŁ"1891.

Myatt, D., P. Torr, S. Nasuto, J. Bishop, and R. Craddock
2002. Napsac: high noise, high dimensional robust estimation. In *Procedings of the British Machine Vision Conference (BMVC'02)*, Pp. 458–467.

Osselman, G., B. Gorte, G. Sithole, and T. Rabbani
2004. Recognising structure in laser scanner point clouds. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, P. 33âŁ"38.

Raguram, R., O. Chum, M. Pollefeys, J. Matas, and J. M. Frahm
2013. Usac: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):2022–2038.

Raguram, R., J.-M. Frahm, and M. Pollefeys
2008. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In *Procedings of the European Conference on Computer Vision. (ECCV'08)*, Pp. 500–513.

Scharstein, D. and R. Szeliski
2003. High-Accuracy Stereo Depth Maps Using Structured Light. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1(June):195–202.

Schnabel, R., R. Wahl, and R. Klein
2007. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.

Torr, P. H. S. and D. Murray
1997. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271âŁ“300.

Torr, P. H. S. and A. Zisserman
2000. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156.

Trung-Thien, T., C. Van-Toan, and L. Denis
2015. Extraction of cylinders and estimation of their parameters from point clouds. *Computers and Graphics*, 46:345–357.