Pattern Recognition 131 (2022) 108805

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Multiple-solutions RANSAC for finding axes of symmetry in fragments of objects

Seyed-Mahdi Nasiri[a], Reshad Hosseini[a,b], Hadi Moradi[a,c,1,*]

[a] School of ECE, College of Engineering, University of Tehran, Tehran, Iran
[b] School of Computer Science, Institute of Research in Fundamental Sciences (IPM), Tehran, Iran
[c] Intelligent Systems Research Institute, SKKU, South Korea

ARTICLE INFO

Article history:
Received 12 November 2020
Revised 30 April 2022
Accepted 20 May 2022
Available online 3 June 2022

Keywords:
Symmetry axis
Multiple-solutions RANSAC
3D Reconstruction

ABSTRACT

The problem of “finding best lines passing through a set of straight lines” has appeared in applications such as archaeological pottery analysis, precision manufacturing, and 3D modelling. In these applications, an instance of this problem is finding the symmetry axis of a symmetrical object from a set of its surface normal lines. We show that the mentioned instance of the problem may have two meaningful local minima, one of which is the symmetry axis, a fact that has been neglected in the literature. A multiple-solutions RANSAC algorithm is proposed for finding initial estimates of both local minima in the presence of outliers. Then, a coordinate-descent algorithm is presented that starts from these initial estimates and finds the local minima of the problem. The proposed coordinate-descent method does not involve any line search procedure, and its convergence is guaranteed. We also provide a proof for the rate of the convergence.

1. Introduction

The symmetry is a property of an object or image that makes it immutable in the face of a transformation [1]. Extracting a symmetry is the pattern recognition problem of finding the repetitive patterns in a 2D image or a 3D object [2]. The world around us is full of all kinds of symmetries. From symmetries in small particles, atoms and cells to the structure of galaxies. In addition, man-made products in engineering and architecture often have a symmetrical design in order to have an optimal structure and production efficiency [1].

The survey papers in [1,3–5] reviewed and classified the variety of methods that extract geometric symmetries. Finding symmetries in 2D and 3D objects is still an active area of research [2,6]. Three main types of symmetries - i.e. reflection symmetry, rotational symmetry, and translational symmetry - have been studied in different researches for different purposes. The reflection symmetry is discussed in various applications such as 3D reconstruction [7,8], model reduction [9], symmetry detection in natural images [10], and choosing good viewpoint for man-made 3D models [11]. Symmetry based model completion methods are proposed in [12,13] which help machine vision system of a harvesting robot to accurately determine the fruits' location.

In the literature of ancient pottery modelling, finding the symmetry (revolution) axis plays an important role in 3D reconstruction of objects and fragments analysis [14–17]. Since ancient pots were usually made on a spinning wheel, their distinctive feature is axial symmetry. So, extracting and using the axis of symmetry is one of the main parts of the process pipelines proposed in the literature of reassembling pottery fragments [18,19]. It is obvious that the surface normals of a radially symmetric object intersect the symmetry axis [20,21]. This leads to the most common method for finding the symmetry axis, which is finding the line passing through a set of normal lines [22]. In the presence of measurement noise, the problem is formulated as a non-linear least-squares optimization problem [23–26].

Some other methods for finding symmetry axis used other lines instead of surface normals [25,27]. Nevertheless, the symmetry axis is again the best line passing through these lines. For example in precision engineering for symmetrical object manufacturing, Angelo et al. in [27] proposed a method for finding the symmetry axis based on the property of complete axially symmetric objects. The property is that any section curve resulting from the intersection of a radially symmetric object with a plane satisfies the reflection symmetry and its symmetry line always intersects the symmetry axis of the object. Hence, the symmetry axis is the best line which passes through all symmetry lines of a set of section curves.

[*] Corresponding author:.
E-mail address: moradih@ut.ac.ir (H. Moradi).
[1] Adjunct research professor

https://doi.org/10.1016/j.patcog.2022.108805
0031-3203/© 2022 Published by Elsevier Ltd.

In the literature of structure from motion and 3D reconstruction, the revolution axis of axial symmetric structures can be used as an extra information for developing new and powerful algorithms. For example, Zhang et al. in [28] utilized the surface normals to find revolution axis of symmetrical structures such as space objects and employed it in a refinement procedure for 3D reconstruction.

In this paper, we study the number of solutions of the nonlinear least-squares problem used for finding the symmetry axis of symmetrical objects. We will show that the optimization problem may have two meaningful distinct local minima, one of which is the symmetry axis. To the best of our knowledge, this has never been addressed in any previous studies.

The main contribution of this paper is proposing a robust algorithm for finding both distinct local minima of the problem. The algorithm consists of

- a new multiple-solutions RANSAC algorithm for finding two initial estimates of the problem solutions, and
- a novel coordinate-descent algorithm for solving the optimization problem starting from these two initial estimates.

The proposed coordinate-descent method has guaranteed convergence, and we present a theorem giving its non-asymptotic rate of convergence.

1.1. Motivation

The problem of “finding the best line passing through a set of lines” is widely used in the literature of the finding the symmetry axis as a solver [17,29,30] or as an initializer [21,27,31].

Angelo et al. in [25] evaluated the most robust methods presented in the literature for finding the symmetry axis [17,20,30,32]. They showed that there are cases where these methods estimate an axis that is very far from the real symmetry axis, and in some instances it is orthogonal to the symmetry axis. They reported the failure rates from 8 to 60 percent for these methods. Most of these methods solve a non-linear least-squares problem to compute the (initial estimate of) symmetry axis. The failure of finding the local minimum of the cost function that corresponds to the symmetry axis can lead to the failure of the algorithms. As we will later see in the experiments, the cost function may have two distinct local minima, one of which is symmetry axis and another is actually orthogonal to the symmetry axis. This is inline with the observations of Angelo et al. in [25].

To prevent this failure, Angelo et al. used the object thickness changes in different directions and search to find the direction of the intersected plane with minimum changes in the thickness. The normal direction of this plane can be used as an initial guess for the direction of symmetry axis. Using this initial estimation can avoid being trapped in the second local minimum. But this method can only be used when both sides of the wall are scanned, and the difference in body thickness changes in different directions is significant. Therefore, an algorithm that is able to find both local minima in the presence of outliers is necessary. The proposed algorithm in this paper can be used by any algorithm, needing an estimate of the symmetry axis, to prevent the failure caused by assuming the second local minimum as the symmetry axis.

2. Preliminaries

2.1. Straight lines: Basics and distance formula

A line $\mathbf{l}$ can be determined by a point $\mathbf{p}$ on the line and a unit direction vector $\mathbf{u}$, and can be represented by $\mathbf{l}(\mathbf{p}, \mathbf{u})$. All points on the line are given by $\mathbf{q}(s) = \mathbf{p} + s\mathbf{u},\ s \in \mathbb{R}$. Given two straight lines, $\mathbf{l}_1(\mathbf{p}_1, \mathbf{u}_1)$ and $\mathbf{l}_2(\mathbf{p}_2, \mathbf{u}_2)$, their distance is the length of a common perpendicular line segment and can be calculated as follows,

$$d(\mathbf{l}_1, \mathbf{l}_2) = \frac{|(\mathbf{p}_2 - \mathbf{p}_1)^T(\mathbf{u}_1 \times \mathbf{u}_2)|}{\|\mathbf{u}_1 \times \mathbf{u}_2\|}. \tag{1}$$

The distance between $\mathbf{l}_1(\mathbf{p}_1, \mathbf{u}_1)$ and an arbitrary point $\mathbf{p}_2$ is obtained by

$$d(\mathbf{l}_1, \mathbf{p}_2) = \|[\mathbf{v}]_\times \mathbf{u}_1\|, \tag{2}$$

where

$$\mathbf{v} = \mathbf{p}_1 - \mathbf{p}_2, \tag{3}$$

and

$$[\mathbf{v}]_\times = \begin{bmatrix} 0 & -\mathbf{v}(z) & \mathbf{v}(y) \\ \mathbf{v}(z) & 0 & -\mathbf{v}(x) \\ -\mathbf{v}(y) & \mathbf{v}(x) & 0 \end{bmatrix}. \tag{4}$$

2.2. Number of lines intersecting given lines

Obviously, there is an infinite number of lines intersecting two given straight lines $\mathbf{l}_1, \mathbf{l}_2$. Such lines can be easily obtained by selecting one arbitrary point on $\mathbf{l}_1$ and another point on $\mathbf{l}_2$. Similarly, there is an infinite number of lines intersecting three given straight lines $\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3$. Consider a plane, P which passes through $\mathbf{l}_1$ and intersects lines $\mathbf{l}_2$ and $\mathbf{l}_3$ at points $\mathbf{p}_2$ and $\mathbf{p}_3$, respectively. A straight line passing through $\mathbf{p}_2$ and $\mathbf{p}_3$ lies on plane P, therefore intersects line $\mathbf{l}_1$. Hence, it passes through all three given lines. Rotating plane P around line $\mathbf{l}_1$ can give an infinite number of such lines. In other words, for every non-parallel three lines there will be infinite number of lines intersecting them.

Adding another line to the set of three non-parallel lines can limit the number of lines passing through these straight lines [33]. solved the problem for four straight lines and showed that, except in degenerate settings, there are two lines that pass through given four lines.

2.3. Evolute

The evolute of a profile is the locus of its centers of curvature. All points on an evolute are the intersection of two normal lines to a profile from infinitesimally adjacent points on the profile. The evolutes of an ellipse and a parabola are shown in Fig. 1. Obviously, if the profile is an arc of a circle, the evolute is a point and all profile normal lines pass through this point. If the profile is not a circle, the evolute is a curve and the normals pass through different points of the evolute.

3. Problem definition & the number of solutions

The optimization problem for finding the best line passing through a set of straight lines is presented in this section. Then, we will explain the number of the solutions of this optimization problem. We will show that for any segment of symmetric structures, the cost function has two distinct local minima, one of which is the symmetry axis and the other gives a point on the evolute of the structure profile.

3.1. The formulation of the optimization problem

In the presence of noise, the problem of finding a line which passes through a set of given lines is reduced to finding a line which has the minimum sum of squared distances from the given lines, i.e.:

$$\mathbf{l}^* = \underset{\mathbf{l}'}{\operatorname{argmin}} \sum_{i=1}^{n} d^2(\mathbf{l}', \mathbf{l}_i). \tag{5}$$

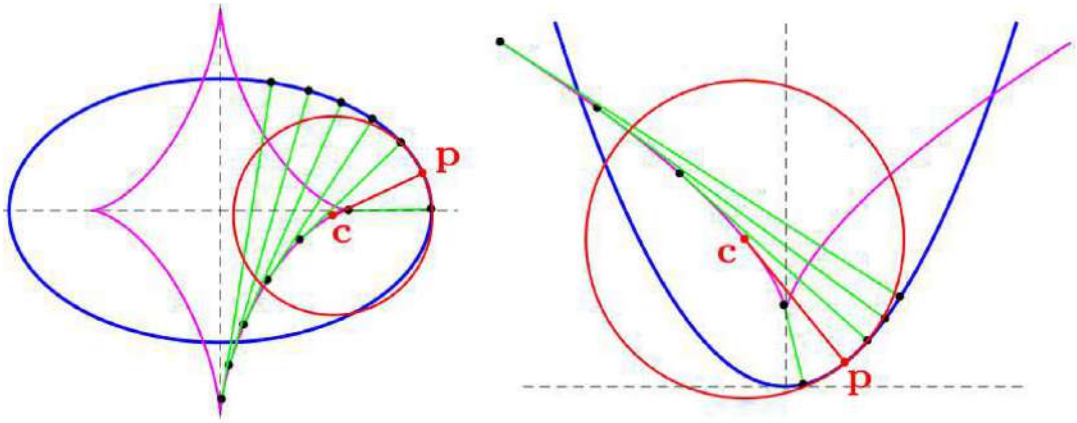

Fig. 1. The evolutes (the magenta curves) of an ellipse and a parabola. Point **c** is the center of curvature at point **p**. The evolute is the locus of all centers of curvature. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

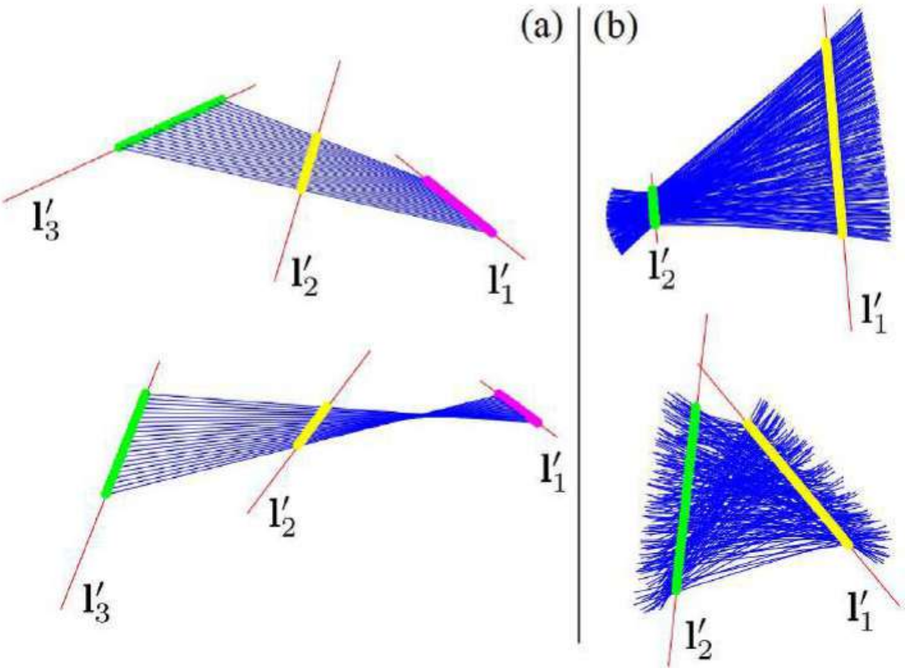

Fig. 2. The red lines pass through all blue lines. (a) There are countless lines, such as $\mathbf{l}'_1$, $\mathbf{l}'_2$ and $\mathbf{l}'_3$, intersecting all blue lines. (b) There are only two lines, $\mathbf{l}'_1$ and $\mathbf{l}'_2$ intersecting all blue lines. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. The number of solutions of the problem

As previously indicated, there are an infinite number of lines that pass through three given lines $\mathbf{l}'_1$, $\mathbf{l}'_2$ and $\mathbf{l}'_3$. Assume that $\mathbf{l}_i$s, $i \in \{1, \ldots, n\}$ are several of these passing lines and suppose that we want to find the best lines passing through $\mathbf{l}_i$s. The sum of squared distances in problem (5), for the lines $\mathbf{l}'_1$, $\mathbf{l}'_2$ and $\mathbf{l}'_3$ are zero, so these are the solutions of the problem. Fig. 2(a) shows this situation from different perspectives. It is clear from the figure and easy to show analytically that in this situation there are other countless lines that pass through $\mathbf{l}_i$s.

Now suppose that $\mathbf{l}'_1$, $\mathbf{l}'_2$ are two straight lines. Assume that $\mathbf{l}_i$s, $i \in \{1, \ldots, n\}$ are arbitrary lines that intersect both $\mathbf{l}'_1$ and $\mathbf{l}'_2$ as shown in Fig. 2(b). In this case, these two lines are the solutions of the problem and there is no other line which passes through all $\mathbf{l}_i$s. Here arises a question, whether this case occurs in practical applications and real experiments.

3.3. The two local minima in real applications

It is obvious that locally estimated surface normals pass through the symmetry axis, a fact that is used in the literature of reconstruction of pottery objects. Since the measurement of normals have noise, the problem of finding the symmetry axis becomes that of finding a line that has the least squares distances from the estimated normal lines as given in (5). In this optimization problem, one of the possible solutions is the symmetry axis. But the problem may have another solution.

As an instance of an axially symmetrical pottery object, a bowl is shown in Fig. 3. Its surface can be obtained by revolving a planar profile around the symmetry axis. Suppose that a part of the profile is an arc of a circle. Thus, for each point on the bowl surface, the radius vector is also the surface normal. Rotating the center of this circle around the symmetry axis, creates circle *c* (dashed magenta circle in Fig. 3). The surface normals pass through both sym-

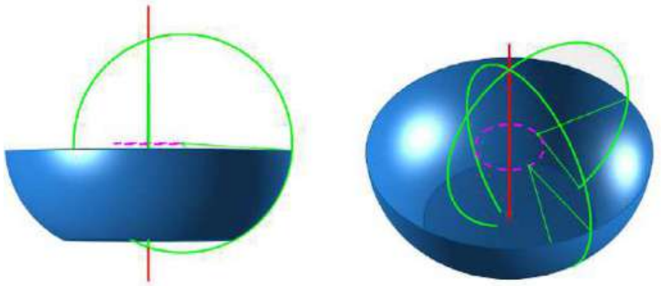

Fig. 3. A bowl and its symmetry axis (red solid line). The bowl wall profile is a part of a circle centered on magenta dashed circle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

metry axis and c. Assume we have a small fragment of the bowl surface. Its local surface normals pass through a small arc of c. If the arc angle is small, it can be approximated by a line. Therefore, this line is another distinct local minimum for the problem (5). In this case, two distinct local minima are found for each fragment of the bowl.

As another example of an axially symmetrical object, consider a dome depicted in Fig. 4. Suppose that an RGB-D Kinect camera is used to scan the dome for constructing its 3D model. As shown in the figure, the surface profile of the dome is an arc of a circle. Each Kinect frame outputs a points cloud and a surface normal line can be found for each point. The symmetry axis can be estimated by finding the line which minimizes the least-squares distances from the normal lines. Similar to the bowl, the least-squares problem has two local minima. Fig. 5 shows a Kinect points cloud, estimated surface normals, and the two local minima.

In general, if we have a set of normals to a profile of a small part of an axially symmetric object, a local minimum of problem (5) is the object's symmetry axis. Assuming noise-free condition, all normals also pass through the evolute curve. Hence, the other local minimum of the problem passes near the evolute. We can now give a short description when the cost function of (5) can have two local minima. If the evolute of a symmetric object is a smooth curve and a fragment of the object is small, then the section of the evolute where the surface normals pass would be a short segment. A short segment of a smooth curve can have a small distance to a straight line. This straight line would be the place of the second local minimum, because distancing infinitesimally from this straight line increases the cost function. We will see in the experiments that this local minimum exists even for large fragments of pottery objects.

In the case of perfect rotational symmetric objects and when the measurement of surface normals do not contain any noise, all surface normals pass through the symmetry axis. Therefore, the axis of rotational symmetry should be the global solution of the problem given in (5), where the cost function becomes zero. In real cases with noisy observations, the normal lines do not pass through the symmetry axis, and therefore the cost value for the symmetry axis is not zero. In addition, in real pottery objects, distortions in the objects' body intensify this phenomenon. Therefore, the value of the cost of the second local minimum can be less than the cost value of the symmetry axis, and therefore the symmetry axis is no longer the global minimum but the second local minimum is the global minimum of the cost function. In the experiments, we will see that this happens in some experiments for real and synthetic datasets.

4. The proposed method

In this part we propose a new *Coordinate Descent with Multiple-Solutions RANSAC (CDMS)* algorithm, which finds the two solutions of problem (5). The proposed method consists of two main parts: a multiple-solutions RANSAC and a coordinate descent algorithm. The method is summarized in Algorithm 1. The two parts are explained in details as follows:

Algorithm 1 Coordinate-Descent Multiple-Solutions (CDMS) algorithm.

```
Input: The number of n straight lines
Output: Two candidates for the line intersecting given lines

Part~1~-~Multiple-solutions RANSAC
$\mathcal{L} \leftarrow \{\}$;
for $t = 1 : n_r$ do
    select 4 lines from $n$ given lines.
    find the two lines, $\mathbf{l}^t_1, \mathbf{l}^t_2$, intersecting 4 selected lines.
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{l}^t_1, \mathbf{l}^t_2\}$
    $NoI(\mathbf{l}^t_j) = n^t_j, j \in \{1, 2\} \leftarrow$ the number of lines that satisfy $d(\mathbf{l}^t_j, \mathbf{l}_i) < d_{th}, i \in \{1, \dots, n\}$.
end for
$\mathbf{l}^*_1 \leftarrow \text{argmax}_{\mathbf{l} \in \mathcal{L}} NoI(\mathbf{l})$
$\mathcal{L}' \leftarrow$ cluster the lines in $\mathcal{L}$ with respect to their distances from $\mathbf{l}^*_1$ and choose the farther cluster.
$\mathbf{l}^*_2 \leftarrow \text{argmax}_{\mathbf{l} \in \mathcal{L}'} NoI(\mathbf{l})$

Part~2~-~Coordinate-descent
for $j = 1 : 2$ do
    $t \leftarrow 0$
    $\mathbf{l}'_t \leftarrow \mathbf{l}^*_j$
    repeat
        if $t < 10$ then
            $\mathcal{I} \leftarrow \{\mathbf{l}_i | d(\mathbf{l}_i, \mathbf{l}'_t) < d_{th}\}$ set of inliers
        end if
        step 1: Find points $\mathbf{q}_i$s which are closest points on the lines $\mathbf{l}_i$s to the line $\mathbf{l}'_t$, where $\mathbf{l}_i \in \mathcal{I}$.
        step 2: Fit a 3D line to $\mathbf{q}_i$s and get it as $\mathbf{l}'_{t+1}$.
        $t \leftarrow t + 1$
    until convergence
    $\mathbf{l}^*_j \leftarrow \mathbf{l}'_t$
end for
```

4.1. The multiple-solutions RANSAC

Since problem (5) is very sensitive to outliers, it is important to use an algorithm to find inliers. The well-known RANSAC algorithm is the method of choice for finding inliers and initial estimates for similar problems. The usual RANSAC algorithm can not be used for our case though, because the problem can have multiple solutions. The multiple hypothesis RANSAC algorithms [34–37] can not be used either, because in this case each solution is related to a portion of data and these algorithms classify the data to estimate the solutions. In our case, similar lines can contribute to both solutions. Therefore in the following, we present our multiple-solutions RANSAC algorithm which finds inlier lines and two initial estimates of the solutions of the problem (5).

In each iteration of the proposed algorithm, four lines are selected randomly from n given lines and the two lines $\mathbf{l}^t_1, \mathbf{l}^t_2$ intersecting those lines are computed using the method explained in [33]. The number of inliers, n^t_1 and n^t_2, will be calculated for

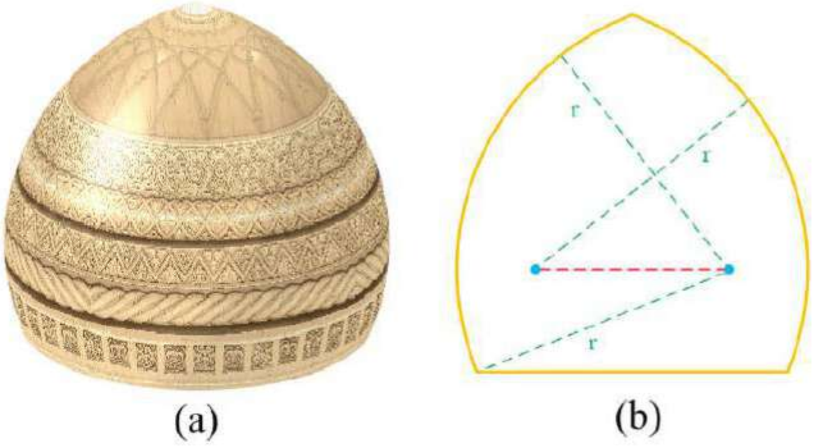

Fig. 4. A dome (a), in which the planar profile (b) is an arc of a circle.

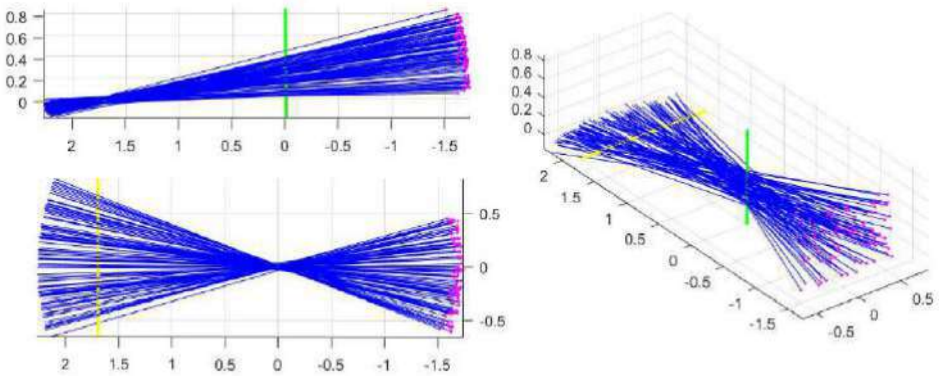

Fig. 5. The surface normals of the dome in Fig. 4 and the two possible solutions of the non-linear least squares problem (5) depicted from different perspectives. The set of magenta points is a points cloud of a part of the dome and the blue lines are the normals. The green and yellow lines are two local minima of the problem. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

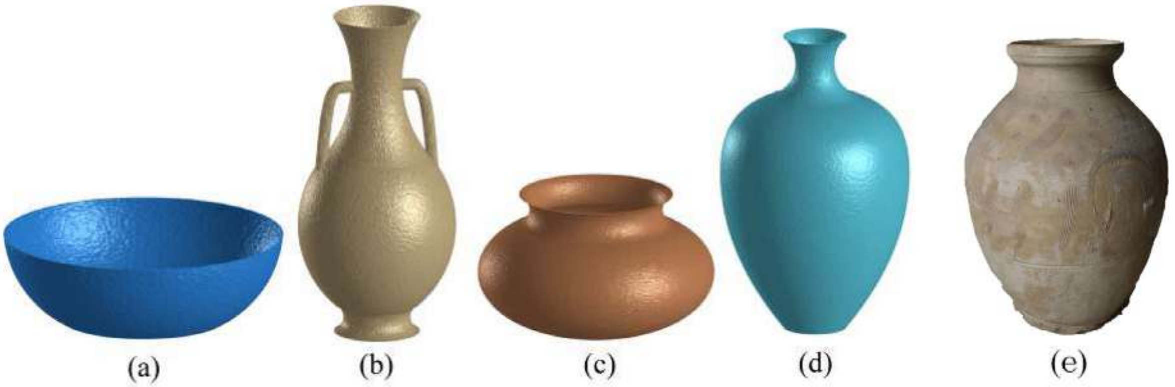

Fig. 6. Four synthetic benchmark datasets: (a) the bowl, (b) the amphora, (c) the flower pot, and (d) the urn, and a real dataset: (e) the earthenware jug. The heights of the objects are 7.7*cm*, 33.8*cm*, 12.5*cm*, 38*cm*, and 46*cm* respectively..

lines $\mathbf{l}_1^t$ and $\mathbf{l}_2^t$. The line $\mathbf{l}_i$ will be considered as an inlier for $\mathbf{l}_j^t$, $j \in \{1, 2\}$ if the distance $d(\mathbf{l}_i, \mathbf{l}_j^t)$ is smaller than a threshold. The threshold can be chosen such that more than about 95% of the correct lines become inliers. This procedure is iteratively repeated for n_r number of iterations. The value of n_r is chosen according to the desired confidence level for the probability of achieving at least one set of four inlier lines. At the end of the procedure we have a set of $2n_r$ lines $\mathcal{L} = \{\mathbf{l}_1^1, \mathbf{l}_2^1, \mathbf{l}_1^2, \mathbf{l}_2^2, \dots, \mathbf{l}_1^{n_r}, \mathbf{l}_2^{n_r}\}$ and the number of inliers for each member of the set ($NoI(\mathbf{l}_j^t) = n_j^t$). The best line which has the maximum number of inliers is selected as one of the initial estimates,

$$\mathbf{l}_1^* = \underset{\mathbf{l} \in \mathcal{L}}{\operatorname{argmin}}\, NoI(\mathbf{l}). \qquad (6)$$

Then, the lines are clustered with respect to their distances from $\mathbf{l}_1^*$ using simple fuzzy c-means clustering [38]. The cluster with far-

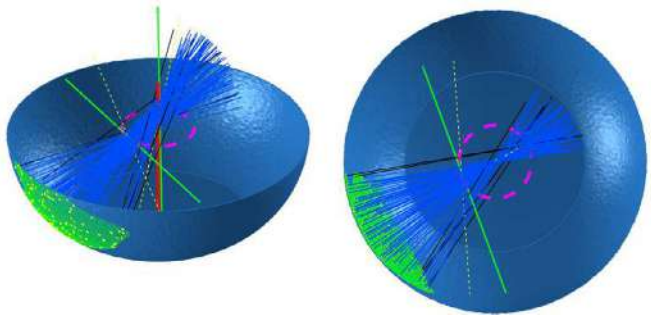

Fig. 7. The two solutions of problem (5) for a fragment of the bowl, i.e the green area. The yellow points on the fragment are the randomly selected points. The blue lines are the normal lines. Dashed yellow lines are $\mathbf{l}_1^*$ and $\mathbf{l}_2^*$, i.e. the outputs of proposed Multiple-solutions RANSAC algorithm. The curvature center of body profile is shown as dashed magenta circle. The final solutions are the green lines. One of them aligned with the symmetry axis and another one is tangent to the dashed magenta circle. The number of 5 outliers are detected for both local minima.. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ther lines is called $\mathcal{L}'$. The other best line is selected from $\mathcal{L}'$:

$$\mathbf{l}_2^* = \operatorname*{argmin}_{\mathbf{l}\in\mathcal{L}'} NoI(\mathbf{l}). \tag{7}$$

Afterward, an iterative algorithm may be used to solve problem (5) for the two sets of inliers. During the optimization, the inliers can be further improved by reselecting the inliers after each iteration of the iterative solver. Although the experimental results show that the inliers do not change after a few iterations, we stop updating them after 10 iterations to maintain the convergence conditions of our coordinate-descent algorithm described in the next subsection.

4.2. The coordinate-descent algorithm

In this section we present a coordinate-descent algorithm to solve the nonlinear least-squares problem (5). The algorithm starts from a line, represented by $\mathbf{l}'_0$, and repeats two following steps:

1. Giving current estimate $\mathbf{l}'_t$, find points $\mathbf{q}_i$s that are closest points on the lines $\mathbf{l}_i$s to $\mathbf{l}'_t$.
2. Fit a 3D line to $\mathbf{q}_i$s and get the new line estimate $\mathbf{l}'_{t+1}$.

Both steps can be computed in closed-form and therefore the algorithm does not need the line-search procedure. Later in the convergence proof, we will show the proposed method is a coordinate-descent algorithm. In the following, the two steps of the algorithm are explained in details.

4.2.1. Step 1

Let's assume $\mathbf{q}_i$ and $\mathbf{q}'_t$ are the closest pair points on the two lines $\mathbf{l}_i$ and $\mathbf{l}'_t$. Therefore line $\mathbf{q}_i - \mathbf{q}'_t$ is perpendicular to both lines. Suppose that $\mathbf{q}_i = \mathbf{p}_i + s_i\mathbf{u}_i$ and $\mathbf{q}'_t = \mathbf{p}'_t + s_t\mathbf{u}'_t$, therefore

$$\mathbf{u}_i^T(\mathbf{q}_i - \mathbf{q}'_t) = \mathbf{u}_i^T(\mathbf{p}_i + s_i\mathbf{u}_i - \mathbf{p}'_t - s_t\mathbf{u}'_t) = 0, \tag{8}$$

$$\mathbf{u}_t'^T(\mathbf{q}_i - \mathbf{q}_t) = \mathbf{u}_t'^T(\mathbf{p}_i + s_i\mathbf{u}_i - \mathbf{p}'_t - s_t\mathbf{u}'_t) = 0. \tag{9}$$

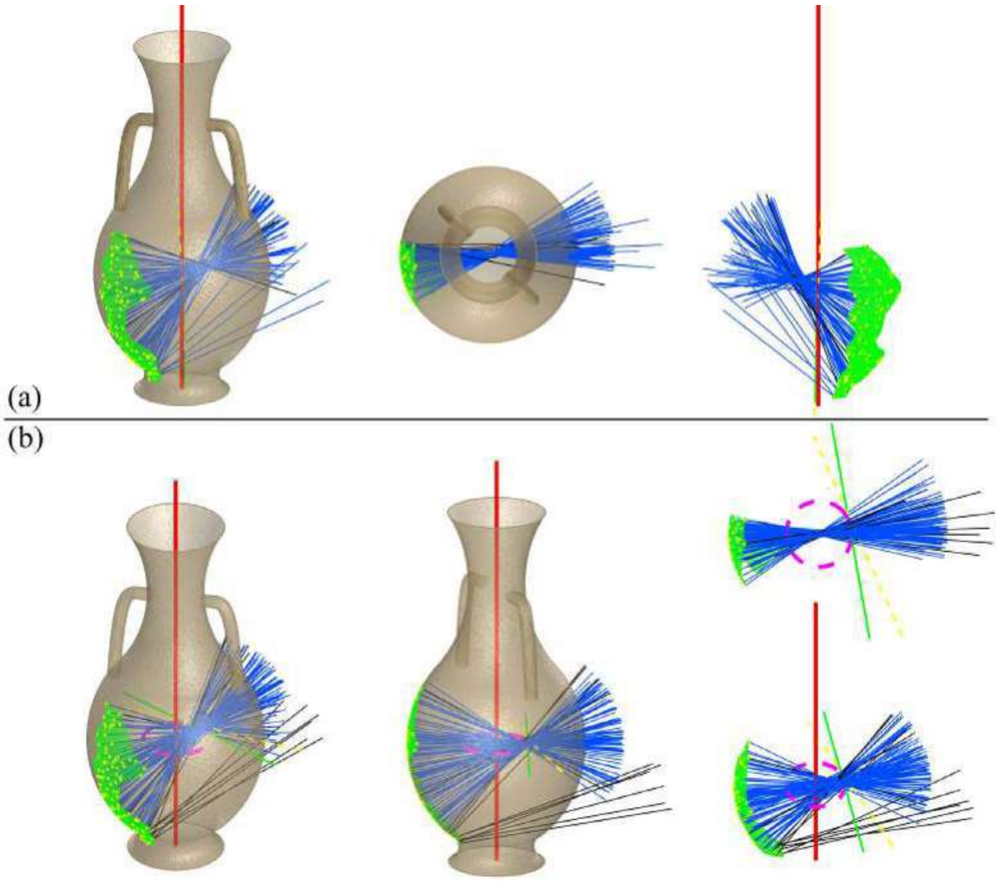

Fig. 8. The selected fragment from the amphora's body (the green points). The description of different curves is the same as that of Fig. 7. Subplots (a) and (b) show the curves for the first and second solutions of the CDMS algorithm. The number of outliers are 0 and 5 for the first and the second local minimum, respectively. . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

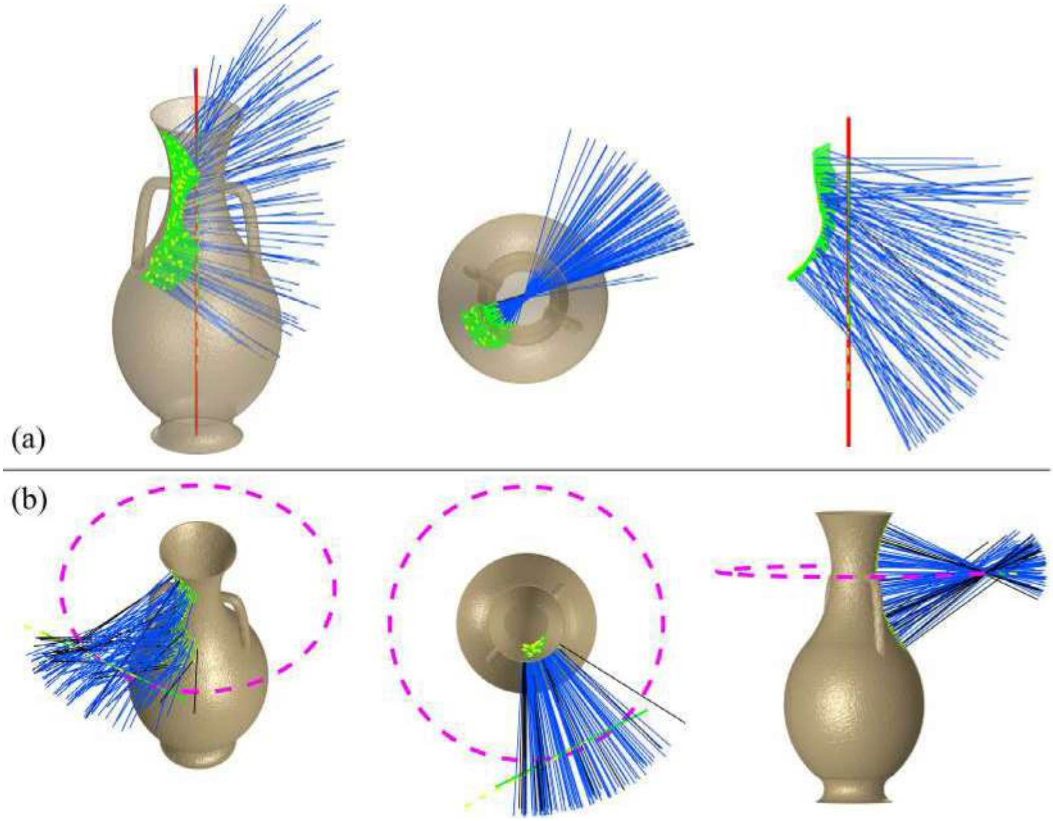

Fig. 9. The lines and the points are the same as Fig. 8 for a fragment of the amphora's neck. The number of 1 and 16 outliers are detected in (a) and (b), respectively.

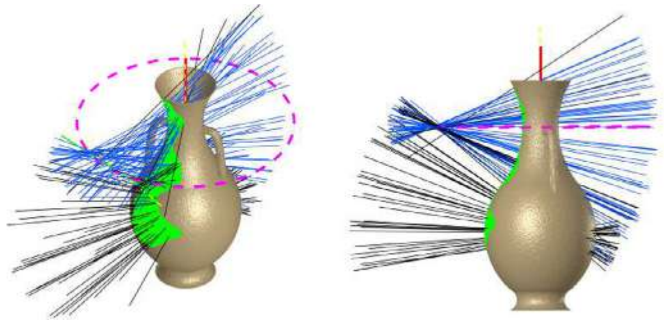

Fig. 10. If a fragment of the amphora contains a part of its body and also a part of its neck, the first solution of the problem (5) is the symmetry axis and the second solution is the center of curvature of the body or the neck.

Solving these linear equations, s_i is obtained as

$$s_i = \frac{(\mathbf{u}_i^T\mathbf{u}'_t)(\mathbf{u}'^T_t(\mathbf{p}_i - \mathbf{p}'_t)) - \mathbf{u}_i^T(\mathbf{p}_i - \mathbf{p}'_t)}{1 - (\mathbf{u}_i^T\mathbf{u}'_t)^2}. \tag{10}$$

Points $\mathbf{q}_i$s, $i \in \mathcal{I}$ are returned as the output of step 1.

4.2.2. Step 2

In this step, a line is fitted on $\mathbf{q}_i$s found in step 1. The best line is the solution of the following least-squares problem,

$$\mathbf{l}'_{t+1} = \underset{\mathbf{l}}{\operatorname{argmin}} \sum_{i\in\mathcal{I}} d(\mathbf{l}, \mathbf{q}_i)^2. \tag{11}$$

It can be shown that, this line pass through the mean of given points. Thus,

$$\mathbf{p}_{t+1} = \operatorname{mean}\{\mathbf{q}_i\}. \tag{12}$$

Defining $\mathbf{v}_i = \mathbf{p}_{t+1} - \mathbf{q}_i$ and using (2), problem (11) is reformulated as,

$$\mathbf{l}'_{t+1} = \underset{\mathbf{u}}{\operatorname{argmin}} \sum_{i\in\mathcal{I}} ([\mathbf{v}_i]_\times \mathbf{u})^T ([\mathbf{v}_i]_\times \mathbf{u}) \tag{13}$$

$$= \underset{\mathbf{u}}{\operatorname{argmin}} \sum_{i\in\mathcal{I}} \mathbf{u}^T [\mathbf{v}_i]_\times^T [\mathbf{v}_i]_\times \mathbf{u} \tag{14}$$

$$= \underset{\mathbf{u}}{\operatorname{argmin}}\, \mathbf{u}^T \left(-\sum_{i\in\mathcal{I}} [\mathbf{v}_i]_\times^2 \right) \mathbf{u} \tag{15}$$

Obviously the optimal solution of (15) is the eigenvector corresponding to the smallest eigenvalue of $\left(-\sum_{i\in\mathcal{I}}[\mathbf{v}_i]_\times^2\right)$.

4.3. Convergence proof

The following theorem shows that the proposed iterative solver is actually a coordinate-descent method and it satisfies the conditions needed for obtaining convergence and even non-asymptotic rate of convergence.

Theorem 1. *Using step 1 and step 2 iteratively, forms a two block coordinate-descent algorithm which minimizes the following over-parametrization version of problem (5).*

$$\{\mathbf{l}^*, \mathbf{q}_1^*, \ldots, \mathbf{q}_n^*\} = \underset{\mathbf{l}, \mathbf{q}_1 \in \mathbf{l}_1, \ldots, \mathbf{q}_n \in \mathbf{l}_n}{\operatorname{argmin}} J(\mathbf{l}, \mathbf{q}_i), \tag{16}$$

where

$$J(\mathbf{l}, \mathbf{q}_i) = \sum_{i=1}^{n} d(\mathbf{l}, \mathbf{q}_i)^2. \tag{17}$$

The algorithm converges to a stationary point. The rate of convergence is $O(1/\sqrt{T})$, where T is the number of iterations.

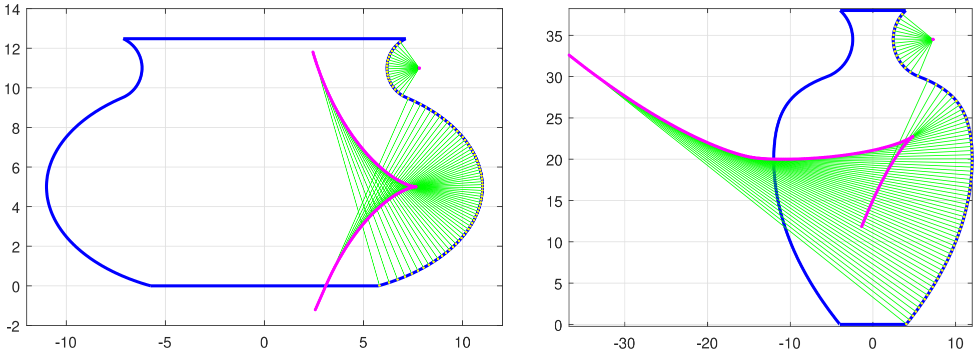

Fig. 11. The evolute of the flower pot (left) and the urn (right). The axes values are in centimeters.

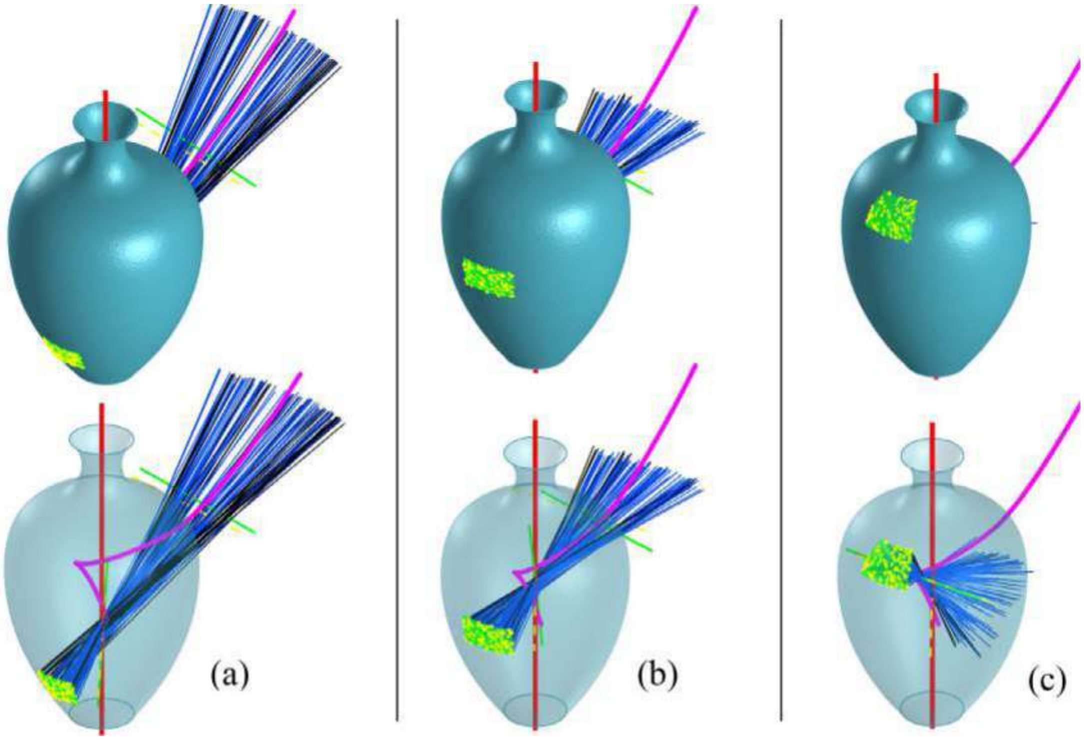

Fig. 12. The points clouds of small fragments (green points) from the bottom (a), middle (b) and top (c) of the urn. The description of different curves is the same as that of Fig. 7, and the magenta curve shows the evolute. One of the solutions aligns to the symmetry axis (the red line) and the other one passes through the evolute. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Proof. Recal that $\mathbf{q}_i = \mathbf{p}_i + s_i\mathbf{u}_i, s_i \in \mathbb{R}$, are the set of all points on $\mathbf{l}_i$, the sum of squared distances in (5) can be re-expressed as $J(\mathbf{l}, s_i) = \sum_{i=1}^{n} d(\mathbf{l}, \mathbf{p}_i + s_i\mathbf{u}_i)^2$. Therefore, it is obvious that a solution of (16) is also a solution of problem (5). Step 1 minimizes $J(\mathbf{l}, s_i)$ over block 's_i' and step 2 minimizes it over block '$\mathbf{l}$'.

Rouzban et al. [39] showed that if the cost function of two block coordinate-descent algorithm satisfies certain conditions, then the algorithm converges and the rate of convergence is at least $O(1/\sqrt{T})$. Following properties show that the proposed coordinate-descent algorithm satisfies the conditions needed for Theorem 1 of [39].

- Step 1 solves the optimization problem with respect to '$\mathbf{l}$' analytically and therefore $\nabla_{\mathbf{l}} J(\mathbf{l}, s_i) = 0$. Furthermore, it reduces the cost function because it finds the minimum of the cost function with respect to this block.
- Step 2 minimizes the cost function with respect to the block 's_i' analytically.
- In step 2, the Hessian of $J(\mathbf{l}, s_i)$ with respect to the s_i block is $\sum_{i=1}^{n} -\mathbf{u}_i[u]_\times[u]_\times\mathbf{u}_i$ and is bounded. Therefore, the gradient of the cost function with respect to the s_i block is Lipschitz continuous. Then using Corollary 3 of [39], the descent condition for Theorem 1 of [39] is satisfied and we get the rate of convergence equal to $O(1/\sqrt{T})$. □

5. Experiments

The performance of the proposed algorithms is evaluated through several experiments. The simulations are two-fold and the results are presented in two subsections. In the first part, we apply the proposed algorithm to a wide variety of fragments of different objects to give the reader an overview of the argued local minima and their positions. So, the results are presented by 3D figures from different perspectives. In the next part, the results are presented in quantitative terms and the results are compared with some well-known algorithms in the literature. The threshold value of the algorithm is set to $d_{th} = .3cm$ in all experiments. This threshold ensures that for almost all experiments, more than 90% of the correct surface normals become inliers[2].

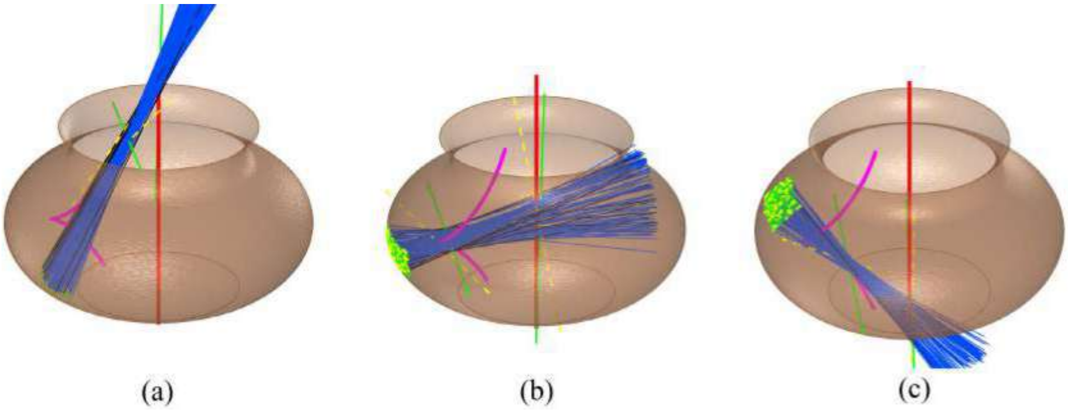

Fig. 13. The points clouds of small fragments (green points) from the bottom (a), middle (b) and top (c) of the flower pot. The description of different curves is the same as that of Fig. 7, and the magenta curve shows the evolute. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

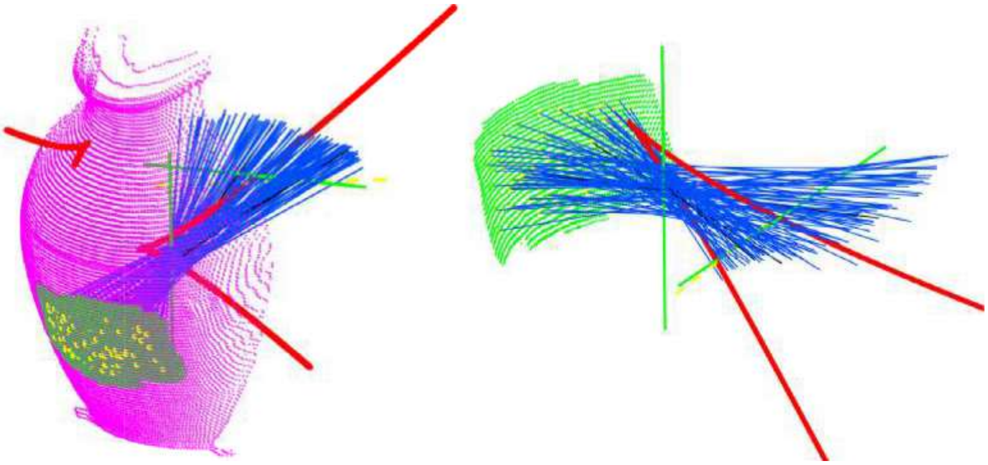

Fig. 14. The points clouds of a fragment (green points) from the bottom part of the jug. The description of different curves is the same as that of Fig. 7, and the magenta curve shows the evolute. The number of outliers are 0 and 3 for the first and the second local minima, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

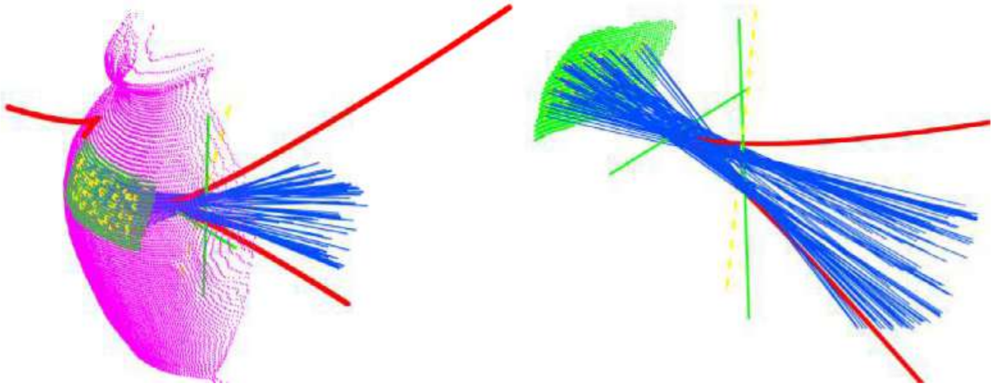

Fig. 15. The results of CDMS algorithm on a fragment of middle part of the jug. The description of different curves is the same as that of Fig. 7, and the magenta curve shows the evolute. The number of outliers are zero for the both local minima. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5.1. Overview and intuitive results

To evaluate the proposed algorithm, four synthetic datasets and a real one are provided. The synthetic datasets are points clouds of axial symmetric objects, i.e. a bowl, a flower pot, an urn, and an amphora, and a real earthenware jug (Fig. 6). We collected the points cloud of the earthenware jug using a Microsoft Kinect V2 scanner. The points are obtained by averaging the depth values over 10 sec from a fixed position. The Kinect depth measurement noise for short distances, i.e. about one meter, is about a few millimeters. Since we have obtained the values by averaging over

[2] To aid reproducing the results, the implementation of the proposed algorithms in MATLAB is available via http://visionlab.ut.ac.ir/resources/cdms.zip.

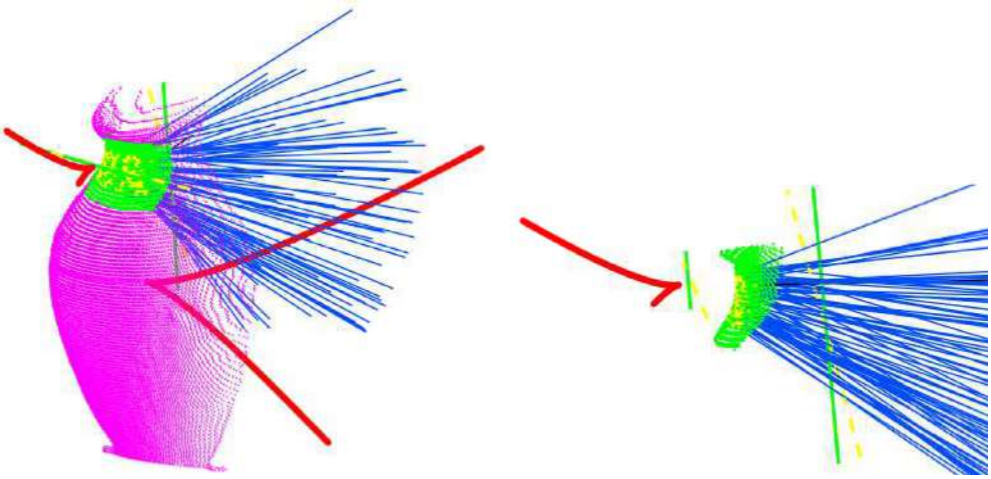

Fig. 16. The results of CDMS algorithm on a wider fragment of the jug. The description of different curves is the same as that of Fig. 7, and the red curve shows the evolute. The number of outliers are 0 and 1 for the first and the second local minimum, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

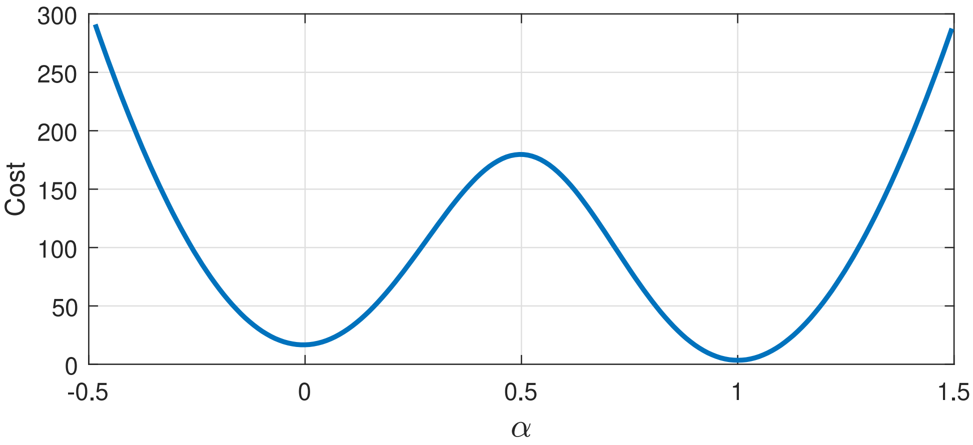

Fig. 17. The cost values for a set of lines $\mathbf{l}_\alpha(\alpha\mathbf{p}_1^* + (1-\alpha)\mathbf{p}_2^*, \alpha\mathbf{u}_1^* + (1-\alpha)\mathbf{u}_2^*)$ as a function of α. The graph is related to the fragment shown in Fig. 14.

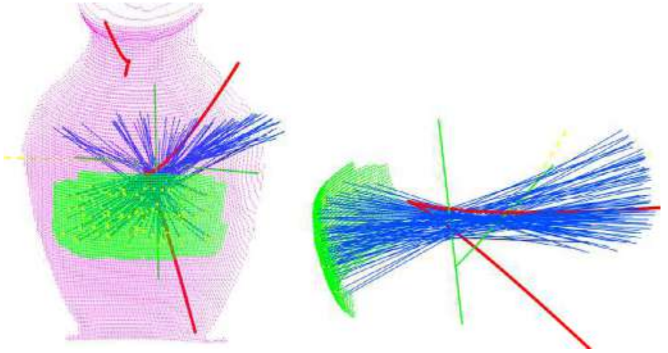

Fig. 18. The results of CDMS algorithm on a fragment of the jug's neck. The description of different curves is the same as that of Fig. 7, and the red curve shows the evolute. The number of outliers are 0 and 1 for the first and the second local minimum, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

time, about 10 seconds, the amount of noise is actually less than this value. The following procedure is performed for each object:

- A fragment of the object is selected.
- Noise is added to the points (except for the jug).
- 100 points were randomly selected.
- The normal vectors were calculated for all of the selected points.
- Problem (5) is solved using the CDMS algorithm.

Fig. 7 shows the selected fragment of the bowl in green. A Gaussian random noise with the standard deviation of 0.1mm is added to the points. The yellow points represents the 100 randomly selected points. For each selected point, a surface is fitted to the points with the distance bellow 5mm from that selected point using the simple least squares method. The normal direction is obtained for each surface. Although using the conventional RANSAC method in the surface fitting step can help to obtain more accurate normals even in the presence of outlier points, this step is skipped to form more outlier normal lines and to evaluate the proposed algorithm in a more challenging scenario. These 100 surface normal lines, that are the input of the proposed RANSAC algorithm, are shown in the figure as blue and black lines. Lines $\mathbf{l}_1^*$ and $\mathbf{l}_2^*$, which are the outputs of the proposed RANSAC algorithm, i.e. lines 8 and 10 of the CDMS algorithm, are shown as dashed yellow lines. The lines which are interpreted as outliers are shown as black lines. The final solutions of the CDMS algorithm are shown as solid green lines. As it can be seen, one of them is close to the symmetry axis of the bowl and the other one is tangent to magenta dashed circle and passes trough center of curvature of the bowl's wall profile.

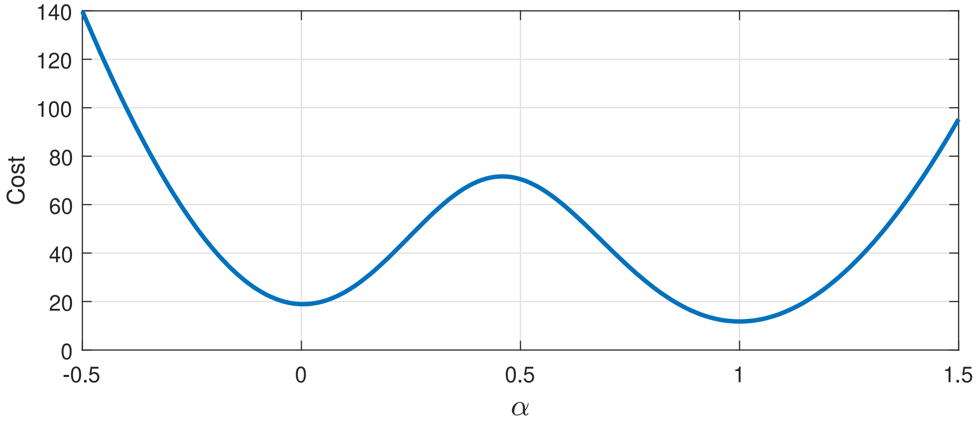

Fig. 19. The cost values for a set of lines $\mathbf{l}_\alpha(\alpha\mathbf{p}_1^* + (1-\alpha)\mathbf{p}_2^*, \alpha\mathbf{u}_1^* + (1-\alpha)\mathbf{u}_2^*)$ as a function of α. The graph is related to the fragment shown in Fig. 18.

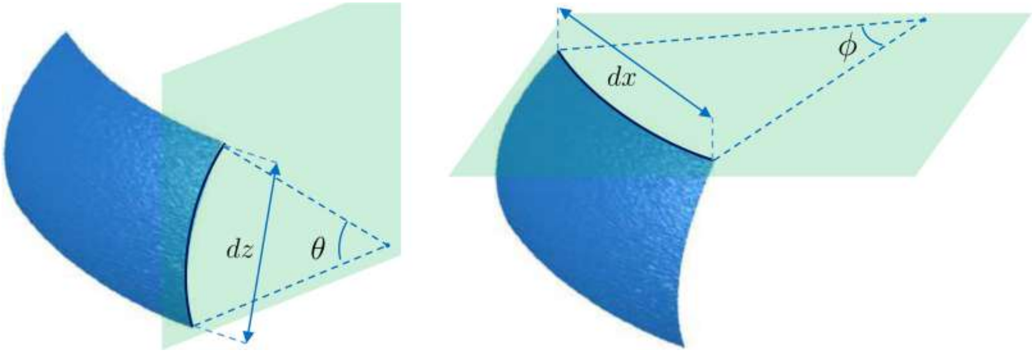

Fig. 20. The geometric characteristics of the fragments are presented by four parameters. Two span angles θ and ϕ, and two lengths dx and dz .

The amphora's profile has different curvatures on its base, body and neck. Fig. 8 shows a points cloud of a fragment which contains a piece of the amphora's body and small piece of the amphora's base. One of the outputs of the proposed RANSAC algorithm and consequently one of the final solutions, shown in Fig. 8(a), is aligned to the amphora's symmetry axis. As shown in the figure, almost all the lines including the amphora's body and the amphora's base normal lines, are detected as inliers by the proposed algorithm. The inliers are shown as blue lines and the black lines are outliers. Fig. 8(b) shows the second result of the proposed algorithm which passes through the center of the curvature of the amphora's body profile. The amphora's base normal lines, that have different curvature profile, are detected as outliers.

The experiment is repeated for a fragment of the amphora's neck. Fig. 9 shows the two results of the proposed algorithm for this fragment. The second solution of the problem for this fragment is located outside of the object, near the magenta circle obtained by rotating the evolute around the symmetry axis.

If the fragment contains both amphora's body and its neck, the fragment shown in Fig. 10, there are three local minima for problem (5). As usual, one local minimum is the symmetry axis. The other two local minima are the line passing through the center of curvature of the amphora's body profile and the line passing through the center of curvature of the amphora's neck. The proposed algorithm returns the symmetry axis as the first solution and one of the other two local minima as the second solution. Note that, these two local minima are related to a subset of the normal lines. For example if the second solution is the line which passes through the amphora's neck, the normals related to the amphora's neck are detected as inliers and the normals related to the amphora's body are detected as outliers. Fig. 10 illustrates this situation. In these cases, common multiple-model RANSAC algorithms, such as [34,36], can be used to classify the lines belonging to the different parts of a fragment. Then, our Multiple-solutions RANSAC algorithm can be used to find both solutions for each part.

The wall profile of the bowl is an arc of a circle, therefore, their evolute is a point. Also, the amphora's profile contains three arcs, and therefore its evolute is a set of three points. The wall profiles of the flower pot and the urn are not an arc of a circle. Their evolute are shown in Fig. 11. As mentioned in Section 2.3, the normal lines of a infinitesimally small fragment of an object passes through its evolute. Thus, the second local minimum of problem (5) for the normal lines of a small fragment is a line which passes near the evolute. Obviously, the locus of all center of curvatures of the flower pot and the urn and any general symmetrical objects are obtained by rotating their evolute, which is a curve, around the symmetry axis. Therefore, the result is a surface. In any case, the concept of the evolute can provide better understanding about approximate location of the second local minimum.

Figs. 12 and 13 show the solutions of the CDMS algorithm for different small fragments of the urn and the flower pot, respectively. Figs. 14, 15 and 16 show the results of the CDMS algorithm on different fragments of the jug. The results of CDMS algorithm on different fragments of the synthetic and the real data with different angular spans are presented in the appendix.

Fig. 17 gives a better understanding about the cost values, i.e. the sum of squared distances from the set of normal lines, of the two local minima of the fragment of jug shown in Fig. 14. The figure depicted the cost values for a set of lines $\mathbf{l}_\alpha(\alpha\mathbf{p}_1^* + (1-\alpha)\mathbf{p}_2^*, \alpha\mathbf{u}_1^* + (1-\alpha)\mathbf{u}_2^*)$ as a function of α, where $\mathbf{l}_1^*(\mathbf{p}_1^*, \mathbf{u}_1^*)$ and $\mathbf{l}_2^*(\mathbf{p}_2^*, \mathbf{u}_2^*)$ are the two local minima. This graph also indicates that a gradient-based algorithm used to solve (5) may converge to a solution which is not the symmetry axis, depending on the start-

Table 1
The comparison of methods in different fragments of the flower pot and the bowl.

Frag. from	Frag. param.	Noise σ(mm)	Method	Fail %	a_c (deg)	d_c (mm)	a_f (deg)	d_f (mm)	Noise σ(mm)	Method	Fail %	a_c (deg)	d_c (mm)	a_f (deg)	d_f (mm)
flower pot	$dx = 9$cm	0.1	CDMS	0	0.9	0.0	−−	−−	0.5	CDMS	0	1.3	0.0	−−	−−
	$dz = 5$cm		PotSAC	0	0.9	0.1	−−	−−		PotSAC	0	1.3	0.0	−−	−−
	$\phi = 70$ deg		BFS	10	0.8	0.0	11.8	0.0		BFS	85	4.6	0.1	15.0	0.1
	$\theta = 70$ deg	0.2	CDMS	0	1.0	0.0	−−	−−	1.0	CDMS	0	3.5	0.1	−−	−−
			PotSAC	0	1.0	0.1	−−	−−		PotSAC	0	3.4	0.1	−−	−−
			BFS	0	1.2	0.1	−−	−−		BFS	100	−−	−−	53.9	26.2
	$dx = 7$cm	0.1	CDMS	0	1.6	0.7	−−	−−	0.5	CDMS	0	2.1	0.2	−−	−−
	$dz = 4$cm		PotSAC	15	1.4	0.7	89.8	71.7		PotSAC	10	2.1	0.2	89.7	71.2
	$\phi = 40$ deg		BFS	0	1.0	0.4	−−	−−		BFS	100	−−	−−	89.6	71.4
	$\theta = 60$ deg	0.2	CDMS	0	1.6	0.7	−−	−−	1.0	CDMS	0	3.5	0.4	−−	−−
			PotSAC	5	1.6	0.7	89.8	71.2		PotSAC	55	3.3	0.3	89.7	71.8
			BFS	10	1.6	0.8	89.7	71.3		BFS	100	−−	−−	89.4	71.7
	$dx = 5$cm	0.1	CDMS	0	0.6	0.2	−−	−−	0.5	CDMS	0	1.8	0.5	−−	−−
	$dz = 3$cm		PotSAC	85	0.2	0.2	90.0	75.3		PotSAC	100	−−	−−	89.9	75.7
	$\phi = 25$ deg		BFS	5	3.7	0.1	90.0	74.8		BFS	100	−−	−−	88.0	67.6
	$\theta = 35$ deg	0.2	CDMS	0	0.6	0.2	−−	−−	1.0	CDMS	5	4.5	1.3	12.1	1.0
			PotSAC	90	0.1	0.3	89.9	75.3		PotSAC	100	−−	−−	89.8	76.9
			BFS	95	0.6	0.3	82.3	67.1		BFS	100	−−	−−	88.0	65.9
bowl	$dx = 13$cm	0.1	CDMS	0	0.4	0.0	−−	−−	0.5	CDMS	0	1.9	0.1	−−	−−
	$dz = 6$cm		PotSAC	0	0.4	0.0	−−	−−		PotSAC	20	1.5	0.2	89.7	58.1
	$\phi = 52$ deg		BFS	0	0.5	0.1	−−	−−		BFS	75	8.0	0.3	55.8	31.3
	$\theta = 40$ deg	0.2	CDMS	0	0.6	0.1	−−	−−	1.0	CDMS	20	7.4	0.3	11.7	0.2
			PotSAC	10	0.3	0.1	89.8	57.8		PotSAC	80	4.3	0.1	79.4	51.6
			BFS	10	7.3	0.1	10.1	0.0		BFS	100	−−	−−	38.4	15.4
	$dx = 9$cm	0.1	CDMS	0	1.3	0.1	−−	−−	0.5	CDMS	0	1.8	0.1	−−	−−
	$dz = 5$cm		PotSAC	45	0.7	0.2	89.3	59.3		PotSAC	20	1.5	0.1	89.2	59.4
	$\phi = 35$ deg		BFS	40	5.0	0.5	31.7	7.5		BFS	40	5.6	0.3	54.3	15.0
	$\theta = 30$ deg	0.2	CDMS	0	3.7	0.1	−−	−−	1.0	CDMS	30	4.8	1.3	14.4	0.3
			PotSAC	65	2.7	0.1	89.3	59.8		PotSAC	85	2.7	0.7	88.3	60.4
			BFS	30	3.9	0.4	14.3	0.1		BFS	100	−−	−−	81.0	56.8
	$dx = 5$cm	0.1	CDMS	0	4.7	0.0	−−	−−	0.5	CDMS	15	3.2	0.4	13.6	0.2
	$dz = 4$cm		PotSAC	90	4.0	0.0	89.4	59.9		PotSAC	80	0.4	0.2	87.0	59.7
	$\phi = 32$ deg		BFS	60	4.3	0.4	71.6	29.7		BFS	100	−−	−−	72.1	6.0
	$\theta = 28$ deg	0.2	CDMS	0	3.6	0.1	−−	−−	1.0	CDMS	30	5.5	1.1	16.5	0.7
			PotSAC	90	4.3	0.1	89.2	59.9		PotSAC	65	3.7	1.6	65.1	43.0
			BFS	85	5.8	0.3	74.0	35.3		BFS	100	−−	−−	65.2	33.8

ing point. The same experiment is repeated for the wider fragment of the jug shown in Fig. 18, and the obtained graph is shown in Fig. 19.

5.2. Quantitative results

In this section the proposed method is evaluated quantitatively, and it is compared to two well-known methods in the literature [22,40]. As mentioned in the introduction, Angelo et al. evaluated several methods of finding the symmetry axis and showed that all of these methods fail in 8 to 60 percent of their experiments [25]. Thus, they proposed to use an object's thickness in different directions to find an initial guess for the direction of symmetry axis. In this method, the change needs to be significant and recognizable in the face of noise and distortion of the objects' body. Therefore, this method is not applicable in our datasets in which the thickness is the same in all directions.

In the following, our method is compared to the other two methods, i.e. PotSAC [22] and BFS [40]. PotSAC comprises two steps: 1) An initialization step done by the MLESAC [41] (a variant of RANSAC) algorithm in which a method based on Pottmann et al. [30] was developed in the paper for minimal axis estimation using a random set of 6 normal lines. 2) A nonlinear refinement step which minimizes the Cao and Mumford's geometric error [20]. BFS is a circle and line fitting method and uses the point cloud of an object, not the normal lines, to find its symmetry axis. In this method a fragment of the object is sliced into many layers along a candidate direction and a circle is fitted onto each layer. Then a line is fitted along the centers of the fitted circles using RANSAC. Finally, the best direction is found through a brute-force search procedure with steps of 5°.

The methods are compared through extensive experiments on a variety of fragments of four synthetic objects in Fig. 6 and four noise levels. The geometric characteristics of the fragments are presented by four parameters shown in Fig. 20. Each experiment is repeated 20 times and the percentage of failure for each method is reported. A failure is when the angle between an estimated symmetry axis and the true symmetry axis is more than 10° or the distance between the two symmetry axes is more than 1*cm*. The results are presented in Tables 1 to 3 in which the following four criteria are reported for the 20 experiments:

- a_c: The mean of the angles between successful estimated axes and the true symmetry axis.
- d_c: The mean of the distances between successful estimated axes and the true symmetry axis.
- a_f: The mean of the angles between failed estimated axes and the true symmetry axis.
- d_f: The mean of the distances between failed estimated axes and the true symmetry axis.

As it can be seen in the tables, CDMS outperforms the other two methods in having lower percentage of failures. In many of the failed cases in PotSAC and BFS, the angle differences between the estimated and true axes are about 90°. Although, CDMS has higher success rates in estimating the symmetry axes, and

Table 2
The comparison of methods in different fragments of the urn.

Frag. from	Frag. param.	Noise σ(mm)	Method	Fail %	a_c (deg)	d_c (mm)	a_f (deg)	d_f (mm)	Noise σ(mm)	Method	Fail %	a_c (deg)	d_c (mm)	a_f (deg)	d_f (mm)
urn - top part	$dx = 7$cm	0.1	CDMS	0	1.7	0.2	−−	−−	0.5	CDMS	0	3.4	0.1	−−	−−
	$dz = 3$cm		PotSAC	30	1.5	0.3	88.0	45.7		PotSAC	50	3.4	0.1	81.4	40.9
	$\phi = 30$ deg		BFS	0	1.9	0.1	−−	−−		BFS	100	−−	−−	79.7	44.6
	$\theta = 20$ deg	0.2	CDMS	0	2.5	0.1	−−	−−	1.0	CDMS	25	3.3	0.5	11.8	0.2
			PotSAC	35	2.1	0.1	88.0	45.7		PotSAC	45	2.3	0.3	86.1	45.4
			BFS	20	4.3	0.0	46.0	21.1		BFS	100	−−	−−	66.8	29.1
	$dx = 9$cm	0.1	CDMS	0	0.8	0.2	−−	−−	0.5	CDMS	0	0.9	0.2	−−	−−
	$dz = 4$cm		PotSAC	5	0.6	0.3	89.4	40.8		PotSAC	10	0.5	0.3	88.2	41.3
	$\phi = 48$ deg		BFS	100	−−	−−	79.7	37.3		BFS	100	−−	−−	86.5	42.3
	$\theta = 35$ deg	0.2	CDMS	0	0.6	0.2	−−	−−	1.0	CDMS	5	2.7	0.1	13.6	0.1
			PotSAC	0	0.5	0.4	−−	−−		PotSAC	25	2.6	0.2	72.1	32.5
			BFS	100	−−	−−	87.0	41.5		BFS	100	−−	−−	85.5	44.4
	$dx = 13$cm	0.1	CDMS	0	0.3	0.0	−−	−−	0.5	CDMS	0	0.7	0.1	−−	−−
	$dz = 6$cm		PotSAC	0	0.2	0.0	−−	−−		PotSAC	0	0.5	0.1	−−	−−
	$\phi = 70$ deg		BFS	0	0.5	0.1	−−	−−		BFS	0	5.5	0.1	−−	−−
	$\theta = 40$ deg	0.2	CDMS	0	0.4	0.0	−−	−−	1.0	CDMS	0	1.8	0.1	−−	−−
			PotSAC	0	0.4	0.0	−−	−−		PotSAC	0	1.8	0.2	−−	−−
			BFS	0	0.4	0.1	−−	−−		BFS	45	8.4	0.2	12.8	0.2
urn - middle part	$dx = 5$cm	0.1	CDMS	0	1.7	0.0	−−	−−	0.5	CDMS	30	4.8	0.3	13.9	0.3
	$dz = 3$cm		PotSAC	0	1.3	0.1	−−	−−		PotSAC	30	2.5	0.5	27.3	17.1
	$\phi = 25$ deg		BFS	0	3.0	0.1	−−	−−		BFS	90	6.0	0.7	37.5	0.6
	$\theta = 15$ deg	0.2	CDMS	0	3.1	0.1	−−	−−	1.0	CDMS	50	4.1	1.5	25.5	1.1
			PotSAC	5	1.9	0.1	89.5	112.7		PotSAC	55	6.5	1.5	50.1	35.7
			BFS	5	3.8	0.3	14.2	0.1		BFS	100	−−	−−	69.9	6.7
	$dx = 9$cm	0.1	CDMS	0	0.6	0.0	−−	−−	0.5	CDMS	0	3.3	0.4	−−	−−
	$dz = 6$cm		PotSAC	0	0.2	0.1	−−	−−		PotSAC	0	1.6	0.8	−−	−−
	$\phi = 40$ deg		BFS	10	3.6	0.1	13.7	0.0		BFS	5	3.3	0.2	10.1	0.3
	$\theta = 25$ deg	0.2	CDMS	0	1.1	0.0	−−	−−	1.0	CDMS	25	6.0	1.4	15.6	0.6
			PotSAC	0	0.3	0.1	−−	−−		PotSAC	20	4.5	2.4	33.9	24.4
			BFS	0	1.8	0.1	−−	−−		BFS	60	5.6	2.1	21.1	0.5
	$dx = 14$cm	0.1	CDMS	0	0.4	0.0	−−	−−	0.5	CDMS	0	1.8	0.1	−−	−−
	$dz = 3$cm		PotSAC	0	0.3	0.0	−−	−−		PotSAC	0	3.3	0.1	−−	−−
	$\phi = 70$ deg		BFS	0	0.8	0.0	−−	−−		BFS	0	2.5	0.1	−−	−−
	$\theta = 10$ deg	0.2	CDMS	0	1.2	0.0	−−	−−	1.0	CDMS	20	4.3	0.5	10.8	0.1
			PotSAC	0	1.3	0.0	−−	−−		PotSAC	30	4.4	0.2	63.5	58.0
			BFS	0	1.3	0.0	−−	−−		BFS	10	4.7	0.2	10.3	0.4

in almost all experiments its estimated axes are the closest to the true ones, PotSAC often estimates the angles better when it succeeds.

6. Conclusion

In this paper, we addressed the problem of “finding the best line passing through a set of straight lines” as a nonlinear least-squares problem. It was shown that, the problem may have two distinct local minima in several practical applications. An algorithm was presented that finds both local minima of the problem in the presence of outliers. To the best of our knowledge, our proposed algorithm is the first work on finding two possible solutions of the problem. The algorithm consists of a Multiple-solutions RANSAC followed by a coordinate-descent algorithm. The presented coordinate-descent algorithm finds the stationary points analytically in each coordinate, and therefore, it is line-search free. We also showed that the algorithm satisfies conditions needed to obtain a rate of convergence.

It was shown that, for a set of normal lines of any small fragment of a symmetrical object, one of the local minima aligns with the symmetry axis of the object. The position of the other local minimum is related to the evolute of the object’s profile. The experiments also showed that two local minima may also exist for large fragments with wider angular spans. Therefore, available gradient-based algorithms may fail to find the symmetry axis, as it was reported in the experiments of [25] where tested methods mistakenly estimated an axis perpendicular to the real symmetry axis. In contrast, the proposed algorithm successfully finds both solutions, the symmetry axis and the other solution that is usually orthogonal to the symmetry axis. The validity of the proposed algorithm was evaluated on synthetic objects and a real scanned object. The proposed algorithm were also compared to two other well-known methods in the literature, and it was shown that the proposed algorithm successfully solves the failure problem of these algorithms.

Our proposed algorithm that is able to find both local minima can be used to develop a method for improving the reconstruction accuracy of symmetrical items. Thus, it has many applications such as archeological object reconstruction and occluded fruit modeling in automatic harvesting. Since the proposed algorithm only uses the normal lines, it can also be used as an initializer for the other methods in which the point cloud is also used to reach better accuracy iteratively, starting from a safe initial axis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 3
The comparison of methods in different fragments of the amphora.

Frag. from	Frag. param.	Noise σ(mm)	Method	Fail %	a_c (deg)	d_c (mm)	a_f (deg)	d_f (mm)	Noise σ(mm)	Method	Fail %	a_c (deg)	d_c (mm)	a_f (deg)	d_f (mm)
amphora	$dx = 2$cm	0.1	CDMS	0	0.3	0.1	––	––	0.5	CDMS	0	1.3	0.2	––	––
- neck	$dz = 3$cm		PotSAC	0	0.2	0.2	––	––		PotSAC	0	1.2	0.2	––	––
	$\phi = 25$ deg		BFS	0	2.8	0.1	––	––		BFS	35	4.8	0.4	13.0	0.2
	$\theta = 15$ deg	0.2	CDMS	0	0.4	0.1	––	––	1.0	CDMS	5	5.6	0.3	12.5	0.1
			PotSAC	0	0.3	0.2	––	––		PotSAC	5	5.5	0.2	12.5	0.1
			BFS	5	3.0	0.1	13.6	0.3		BFS	100	––	––	70.4	4.7
	$dx = 2$cm	0.1	CDMS	0	0.3	0.0	––	––	0.5	CDMS	0	0.8	0.2	––	––
	$dz = 7$cm		PotSAC	0	0.1	0.4	––	––		PotSAC	0	0.6	0.3	––	––
	$\phi = 25$ deg		BFS	15	2.0	0.4	80.7	65.1		BFS	5	2.8	0.5	11.3	0.1
	$\theta = 30$ deg	0.2	CDMS	0	0.2	0.1	––	––	1.0	CDMS	0	1.9	0.4	––	––
			PotSAC	0	0.2	0.3	––	––		PotSAC	20	1.2	0.6	58.9	71.1
			BFS	25	1.5	0.3	79.7	38.0		BFS	90	3.2	0.4	84.0	21.4
	$dx = 2$cm	0.1	CDMS	0	0.1	0.1	––	––	0.5	CDMS	0	0.2	0.3	––	––
	$dz = 10$cm		PotSAC	0	0.0	0.6	––	––		PotSAC	0	0.1	0.8	––	––
	$\phi = 25$ deg		BFS	0	0.2	0.0	––	––		BFS	0	1.5	0.2	––	––
	$\theta = 40$ deg	0.2	CDMS	0	0.1	0.1	––	––	1.0	CDMS	0	0.6	0.5	––	––
			PotSAC	0	0.0	0.5	––	––		PotSAC	10	0.4	1.0	89.7	156.7
			BFS	0	0.4	0.1	––	––		BFS	95	0.9	1.9	87.3	115.7
amphora	$dx = 4$cm	0.1	CDMS	0	1.1	0.0	––	––	0.5	CDMS	0	2.6	0.1	––	––
- body	$dz = 6$cm		PotSAC	0	0.4	0.1	––	––		PotSAC	5	1.7	0.1	89.7	29.1
	$\phi = 22$ deg		BFS	5	2.1	0.1	14.8	0.1		BFS	20	3.3	0.1	23.0	0.2
	$\theta = 27$ deg	0.2	CDMS	0	2.7	0.2	––	––	1.0	CDMS	35	5.1	1.1	12.8	0.2
			PotSAC	0	1.7	0.4	––	––		PotSAC	45	3.0	1.5	55.6	14.2
			BFS	0	2.9	0.2	––	––		BFS	85	8.0	1.1	55.6	11.8
	$dx = 4$cm	0.1	CDMS	0	0.9	0.1	––	––	0.5	CDMS	5	3.6	0.3	11.6	0.2
	$dz = 10$cm		PotSAC	15	0.3	0.2	89.3	29.8		PotSAC	45	2.3	0.5	80.5	26.1
	$\phi = 30$ deg		BFS	5	0.8	0.1	89.8	30.0		BFS	45	4.4	1.0	23.3	0.4
	$\theta = 60$ deg	0.2	CDMS	0	1.4	0.1	––	––	1.0	CDMS	45	4.5	1.2	16.5	0.7
			PotSAC	25	1.0	0.1	89.6	29.7		PotSAC	75	3.7	1.1	66.1	19.3
			BFS	0	3.3	0.2	––	––		BFS	100	––	––	85.5	26.9
	$dx = 7$cm	0.1	CDMS	0	0.2	0.0	––	––	0.5	CDMS	0	1.0	0.1	––	––
	$dz = 9$cm		PotSAC	0	0.1	0.0	––	––		PotSAC	0	0.9	0.1	––	––
	$\phi = 65$ deg		BFS	0	0.4	0.0	––	––		BFS	0	1.0	0.1	––	––
	$\theta = 35$ deg	0.2	CDMS	0	0.3	0.0	––	––	1.0	CDMS	0	5.0	0.1	––	––
			PotSAC	0	0.2	0.0	––	––		PotSAC	0	4.7	0.2	––	––
			BFS	0	0.7	0.1	––	––		BFS	60	2.1	0.4	84.0	24.4

Appendix A. Applying CDMS algorithm to fragments with different angular spans

In this section we apply the CDMS algorithm to fragments with different angular spans. Also the cost values of the set of lines, $\mathbf{l}_\alpha$, are presented for each fragment (as shown in Fig. 17). The set of lines $\{\mathbf{l}_\alpha | \alpha \in \{-.5, -.4, \ldots, 1.4, 1.5\}\}$ for a fragment of the jug are shown in Fig. 21. As explained in the paper, the line $\mathbf{l}_\alpha(\mathbf{p}_\alpha, \mathbf{u}_\alpha) \equiv \mathbf{l}_\alpha(\alpha\mathbf{p}_1^* + (1-\alpha)\mathbf{p}_2^*, \alpha\mathbf{u}_1^* + (1-\alpha)\mathbf{u}_2^*)$, where $\mathbf{l}_1^*(\mathbf{p}_1^*, \mathbf{u}_1^*)$ and $\mathbf{l}_2^*(\mathbf{p}_2^*, \mathbf{u}_2^*)$ are the solutions of the CDMS algorithm. Obviously, the lines $\mathbf{l}_0$ and $\mathbf{l}_1$ are matched to the lines $\mathbf{l}_2^*$ and $\mathbf{l}_1^*$ respectively. Figs. 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36 shows the results of the CDMS algorithm and cost of the set of $\mathbf{l}_\alpha$ lines for different fragments of datasets with different angular spans.

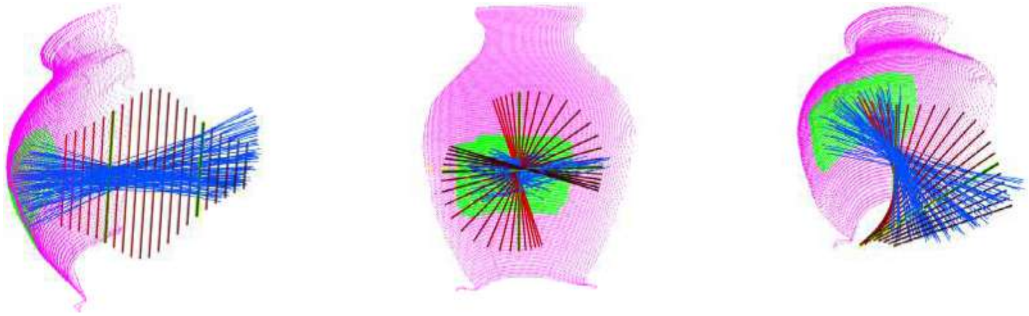

Fig. 21. The green lines are the results of CDMS algorithm, i.e. $\mathbf{l}_1^*$ and $\mathbf{l}_2^*$. The red lines are the set of lines $\{\mathbf{l}_\alpha | \alpha \in \{-0.5, -0.4, \ldots, 1.4, 1.5\}\}$. The lines from dark red to light red correspond to $\alpha = -0.5$ to $\alpha = +1.5$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

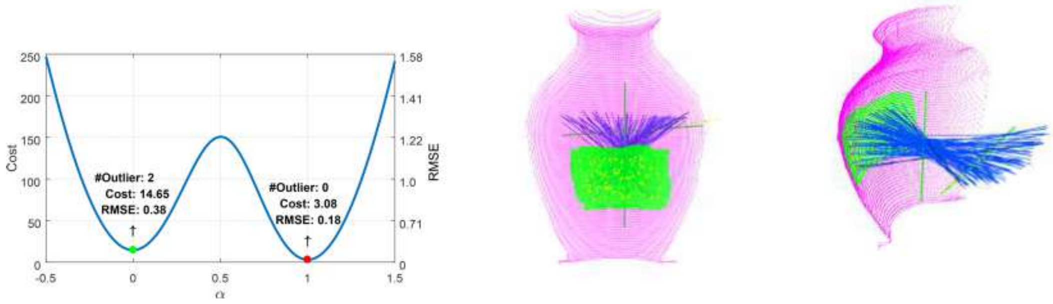

Fig. 22. The results for a fragment of the jug with angular span of 50 degrees.

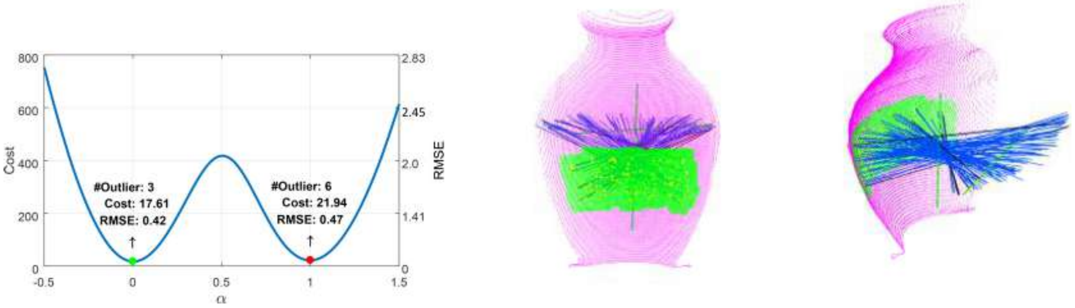

Fig. 23. The results for a fragment of the jug with angular span of 90 degrees.

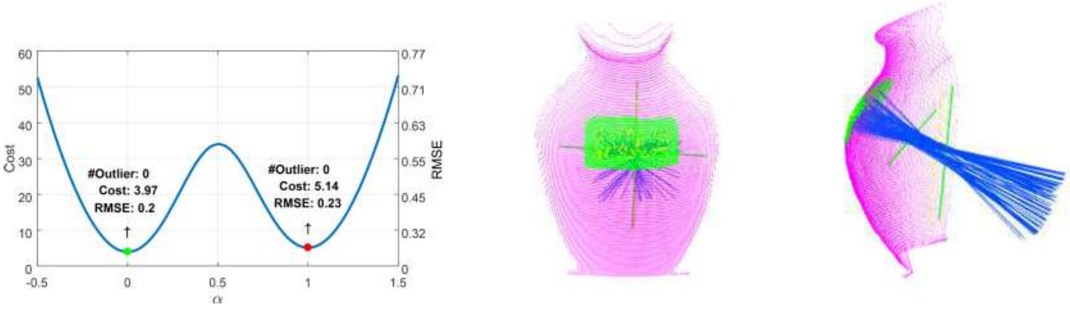

Fig. 24. The results for a fragment from a middle part of the jug with angular span of 50 degrees.

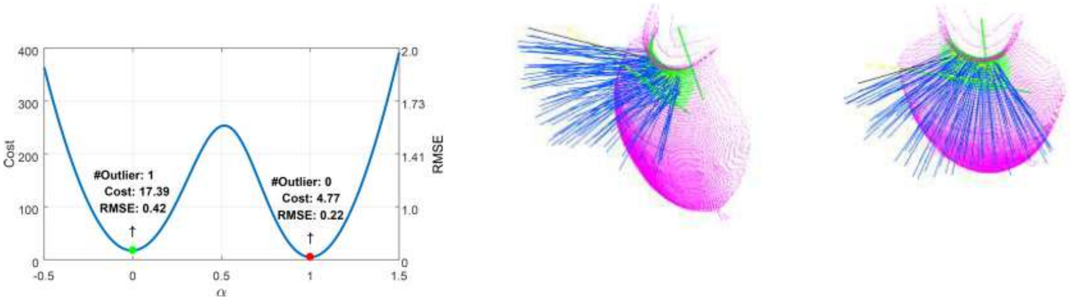

Fig. 25. The results for a fragment from the jug's neck with angular span of 80 degrees.

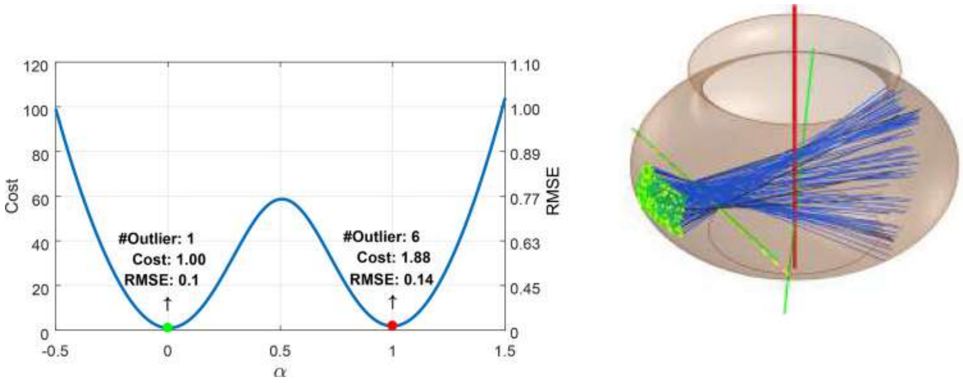

Fig. 26. The results for a fragment from the flower pot with angular span of 30 degrees.

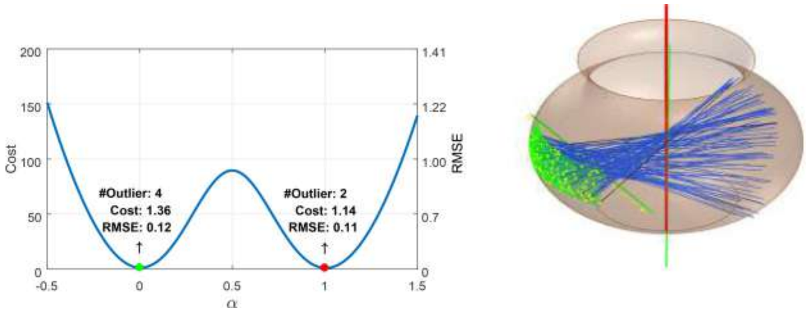

Fig. 27. The results for a fragment from the flower pot with angular span of 60 degrees.

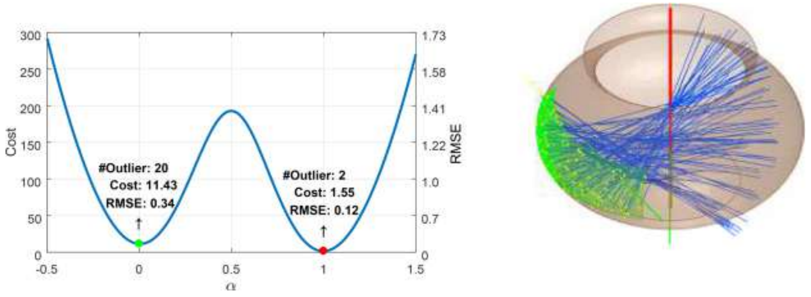

Fig. 28. The results for a fragment from the flower pot with angular span of 90 degrees.

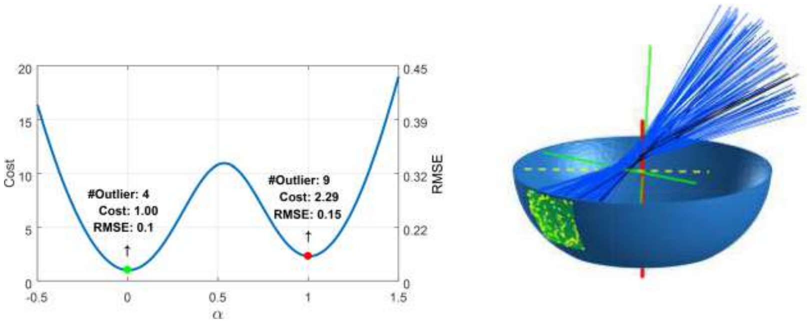

Fig. 29. The results for a fragment from the bowl with angular span of 30 degrees.

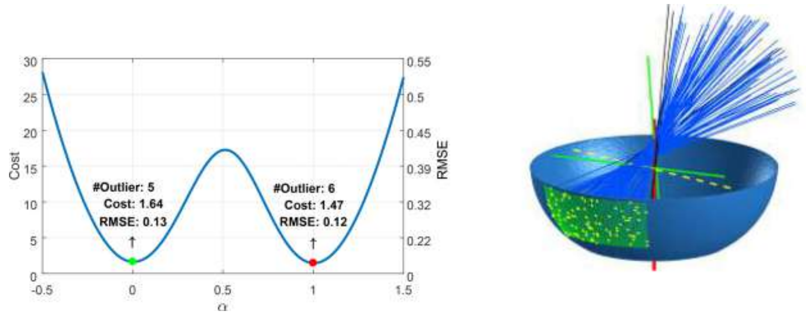

Fig. 30. The results for a fragment from the bowl with angular span of 60 degrees.

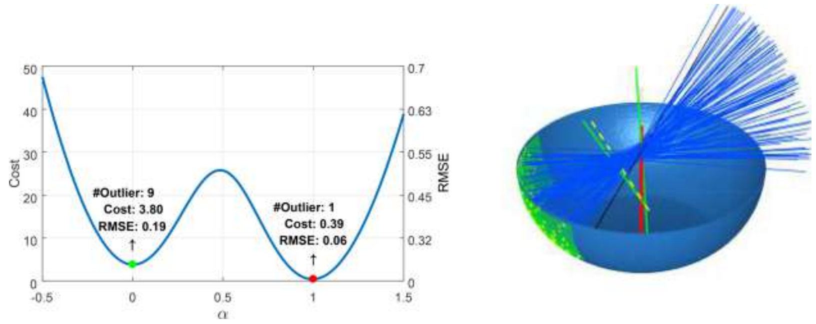

Fig. 31. The results for a fragment from the bowl with angular span of 90 degrees.

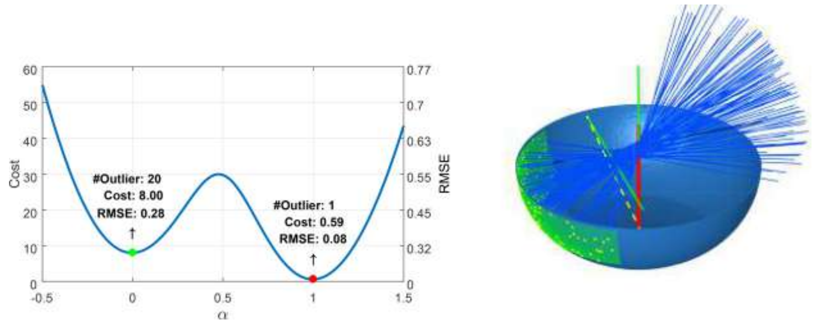

Fig. 32. The results for a fragment from the bowl with angular span of 120 degrees.

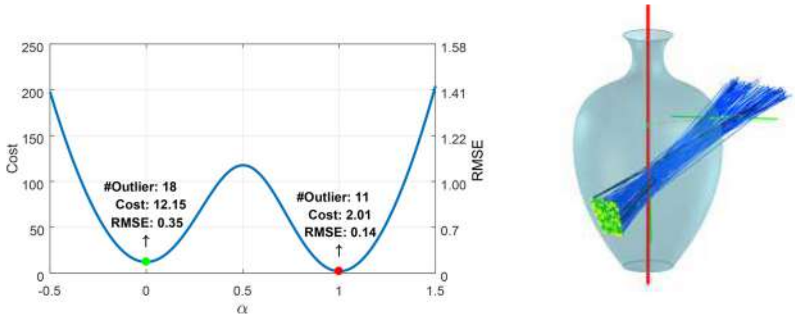

Fig. 33. The results for a fragment from the urn with angular span of 30 degrees.

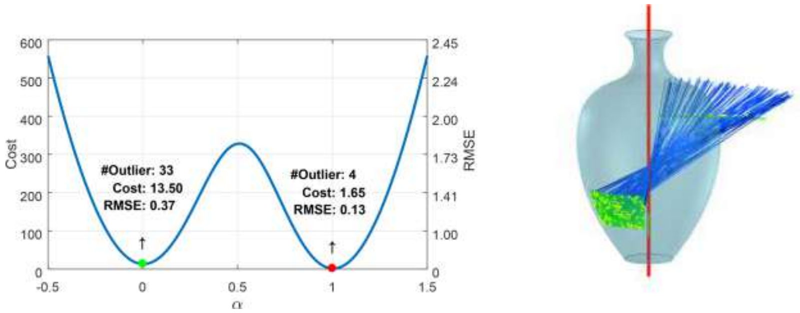

Fig. 34. The results for a fragment from the urn with angular span of 60 degrees.

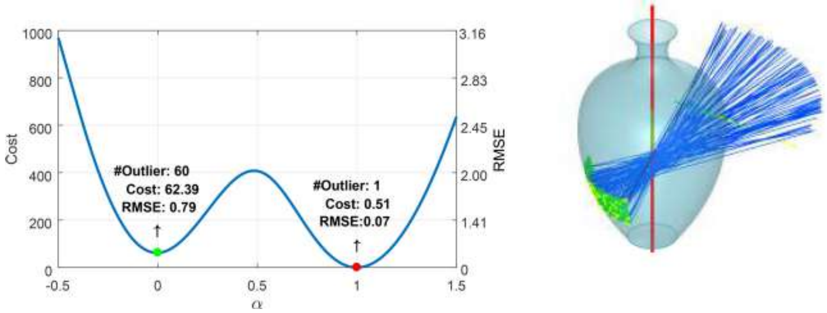

Fig. 35. The results for a fragment from the urn with angular span of 90 degrees.

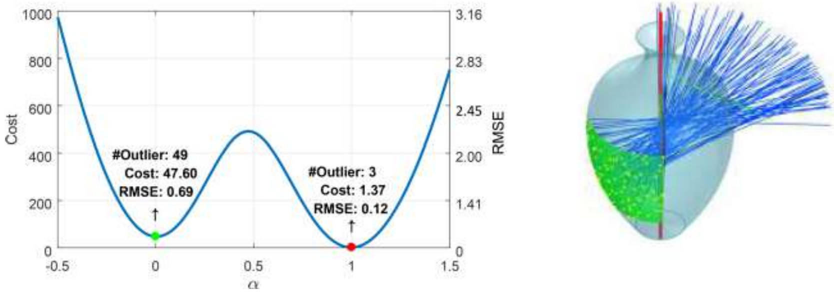

Fig. 36. The results for a big fragment from the urn with angular span of 90 degrees.

References

[1] N.J. Mitra, M. Pauly, M. Wand, D. Ceylan, Symmetry in 3D geometry: Extraction and applications, in: Computer Graphics Forum, volume 32, Wiley Online Library, 2013, pp. 1–23.

[2] R. Nagar, S. Raman, 3DSymm: robust and accurate 3D reflection symmetry detection, Pattern Recognit 107 (2020) 107483.

[3] E. Guardiani, A. Morabito, An investigation on methods for axis detection of high-density generic axially symmetric mechanical surfaces for automatic geometric inspection, Institution of Mechanical Engineers 235 (5) (2021) 920–933.

[4] D. Eslami, L. Di Angelo, P. Di Stefano, C. Pane, Review of computer-based methods for archaeological ceramic sherds reconstruction, Virtual Archaeology Review 11 (23) (2020) 34–49.

[5] Y. Liu, H. Hel-Or, C.S. Kaplan, Computational symmetry in computer vision and computer graphics, Now Publishers Inc, Hanover, MA 02339, USA, 2010.

[6] W. Aguilar, M. Alvarado-Gonzalez, E. Garduño, C. Velarde, E. Bribiesca, Detection of rotational symmetry in curves represented by the slope chain code, Pattern Recognit 107 (2020) 107421.

[7] P. Speciale, M.R. Oswald, A. Cohen, M. Pollefeys, A symmetry prior for convex variational 3D reconstruction, in: European Conference on Computer Vision, Springer, 2016, pp. 313–328.

[8] S.N. Sinha, K. Ramnath, R. Szeliski, Detecting and reconstructing 3D mirror symmetric objects, in: European Conference on Computer Vision, Springer, 2012, pp. 586–600.

[9] N.J. Mitra, L.J. Guibas, M. Pauly, Partial and approximate symmetry detection for 3D geometry, ACM Trans Graph 25 (2006) 560–568.

[10] W. Shen, X. Bai, Z. Hu, Z. Zhang, Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images, Pattern Recognit 52 (2016) 306–316.

[11] C.-K. Lin, W.-K. Tai, Automatic upright orientation and good view recognition for 3D man-made models, Pattern Recognit 45 (2012) 1524–1530.

[12] Y. Ge, Y. Xiong, P.J. From, Symmetry-based 3D shape completion for fruit localisation for harvesting robots, Biosyst. Eng. 197 (2020) 188–202.

[13] S. Thrun, B. Wegbreit, Shape from symmetry, in: IEEE International Conference on Computer Vision, volume 2, 2005, pp. 1824–1831.

[14] J. Wilczek, F. Monna, A. Jébrane, C.L. Chazal, N. Navarro, S. Couette, C.C. Smith, Computer-assisted orientation and drawing of archaeological pottery, Journal on Computing and Cultural Heritage 11 (4) (2018) 1–17.

[15] J.E. Kim, M.Z. Arshad, Y.S. Jong, J.-H. Hong, J. Kim, Y.M. Kim, 3D pots configuration system by optimizing over geometric constraints, in: International Conference on Pattern Recognition, IEEE, 2021, pp. 2398–2405.

[16] C. Liu, W. Hu, Real-time geometric fitting and pose estimation for surface of revolution, Pattern Recognit 85 (2019) 90–108.

[17] R. Halir, An automatic estimation of the axis of rotation of fragments of archaeological pottery: a multi-step model-based approach, in: International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media, 1999.

[18] J.H. Hong, S.J. Yoo, M.A. Zeeshan, Y.M. Kim, J. Kim, Structure-from-sherds: Incremental 3D reassembly of axially symmetric pots from unordered and mixed fragment collections, in: IEEE International Conference on Computer Vision, 2021, pp. 5443–5451.

[19] L. Di Angelo, P. Di Stefano, E. Guardiani, A.E. Morabito, A 3D informational database for automatic archiving of archaeological pottery finds, Sensors 21 (3) (2021) 978.

[20] Y. Cao, D. Mumford, Geometric structure estimation of axially symmetric pots from small fragments, in: International Conference on Signal Processing, Pattern Recognition, and Applications, 2002, pp. 92–99.

[21] A. Willis, X. Orriols, D.B. Cooper, Accurately estimating sherd 3D surface geometry with application to pot reconstruction, in: IEEE Workshop on Applications of Computer Vision in Archaeology, held in conjunction with IEEE Conference on Computer Vision and Pattern Recognition, 2003.

[22] J.H. Hong, Y.M. Kim, K.-C. Wi, J. Kim, PotSAC: A robust axis estimator for axially symmetric pot fragments, in: IEEE International Conference on Computer Vision Workshops, 2019, pp. 1421–1428.

[23] M. Kampel, R. Sablatnig, On 3D modelling of archaeological sherds, in: International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, 1999, pp. 95–98.

[24] A. Karasik, U. Smilansky, 3D scanning technology as a standard archaeological tool for pottery analysis: practice and theory, J. Archaeol. Sci. 35 (2008) 1148–1168.

[25] L. Di Angelo, P. Di Stefano, Axis estimation of thin-walled axially symmetric solids, Pattern Recognit Lett 106 (2018) 47–52.

[26] L. Di Angelo, P. Di Stefano, C. Pane, An automatic method for pottery fragments analysis, Measurement 128 (2018) 138–148.

[27] L. Di Angelo, P. Di Stefano, A.E. Morabito, A robust method for axis identification, Precis. Eng. 39 (2015) 194–203.

[28] H. Zhang, Q. Wei, Z. Jiang, 3D reconstruction of space objects from multi-views by a visible sensor, Sensors 17 (2017) 1689.

[29] S.B. Yacoub, C. Menard, Robust axis determination for rotational symmetric objects out of range data, in: 21th Workshop of the Oeagm, Hallstatt, Austria, 1997, pp. 197–202.

[30] H. Pottmann, M. Peternell, B. Ravani, An introduction to line geometry with applications, Comput.-Aided Des. 31 (1999) 3–16.

[31] M. Kampel, H. Mara, R. Sablatnig, Robust 3D reconstruction of archaeological pottery based on concentric circular rills, in: 6th International Workshop on Image Analysis for Multimedia Interactive Services, Montreux, Switzerland, 2005, pp. 14–20.

[32] D. Han, H.-S. Hahn, Axis estimation and grouping of rotationally symmetric object segments, Pattern Recognit 47 (2014) 296–312.

[33] A. Bieliński, C. Łapińska, Construction of straight line intersecting four given straight lines, Journal Biuletyn of Polish Society for Geometry and Engineering Graphics 18 (2008) 5–8.

[34] P.C. Niedfeldt, R.W. Beard, Recursive RANSAC: multiple signal estimation with outliers, in: IFAC Symposium on Nonlinear Control Systems, volume 46, 2013, pp. 430–435.

[35] P.C. Niedfeldt, Recursive-RANSAC: A novel algorithm for tracking multiple targets in clutter, Brigham Young University-Provo, 2014 Ph.D. thesis.

[36] S.-W. Yang, C.-C. Wang, Multiple-model RANSAC for ego-motion estimation in highly dynamic environments, in: IEEE International Conference on Robotics and Automation, 2009, pp. 389–396.

[37] O. Gallo, R. Manduchi, A. Rafii, CC-RANSAC: Fitting planes in the presence of multiple surfaces in range data, Pattern Recognit Lett 32 (2011) 403–410.

[38] J.C. Bezdek, Pattern recognition with fuzzy objective function algorithms, Springer Science & Business Media, 2013.

[39] S.M. Rouzban, R. Hosseini, A rate of convergence for two-block coordinate descent, arXiv preprint arXiv:1901.08794 (2019).

[40] K. Son, E.B. Almeida, D.B. Cooper, Axially symmetric 3D pots configuration system using axis of symmetry and break curve, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 257–264.

[41] P.H. Torr, A. Zisserman, Mlesac: a new robust estimator with application to estimating image geometry, Comput. Vision Image Understanding 78 (1) (2000) 138–156.

Seyed-Mahdi Nasiri received the B.Sc. degree in electrical engineering from Shahed University, in 2010 and received the M.Sc. degree in control engineering from the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran, in 2012. He is currently working toward the Ph.D. degree in machine intelligence and robotics with the School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran. From 2008 to 2012, he was a member of Omid robotics team and received several international competitions rewards.

Reshad Hosseini received the B.Sc. degree in electrical engineering from the School of Electrical and Computer Engineering, University of Tehran, in 2004, and the Ph.D. degree from the Faculty of Elektrotechnik und Informatik, TU Berlin, Berlin, Germany, in 2012. He is currently an Assistant Professor with the School of Electrical and Computer Engineering, University of Tehran. His current research interests include manifold optimization, largescale mixture models, 3-D reconstruction, neural system identification and deep learning for recognition.

Hadi Moradi received the B.Sc. degree in electrical engineering from the School of Electrical and Computer Engineering, University of Tehran, in 1988, and the Ph.D. degree in computer engineering from University of Southern California, Los Angeles, USA, in 2012. He is currently an Associate Professor with the School of Electrical and Computer Engineering, University of Tehran. His current research interests include robotics, cognitive games, intelligent autism screening and rehabilitation