Published in final edited form as:
Comput Aided Des. 2009 April 1; 41(4): 293–305. doi:10.1016/j.cad.2008.10.012.

Robust principal axes determination for point-based shapes using least median of squares

Yu-Shen Liu[a,*] and **Karthik Ramani**[a,b]

[a]School of Mechanical Engineering, Purdue University, West Lafayette, IN, 47907, USA

[b]School of Electrical Computer Engineering (by courtesy), Purdue University, West Lafayette, IN, 47907, USA

Abstract

A robust technique for determining the principal axes of a 3D shape represented by a point set, possibly with noise, is presented. We use techniques from robust statistics to guide the classical principal component analysis (PCA) computation. Our algorithm is based on a robust statistics method: least median of squares (LMS), for outlier detection. Using this method, an outlier-free major region of the shape is extracted, which ignores the effect on other minor regions regarded as the outliers of the shape.

In order to effectively approximate the LMS optimization, the forward search technique is utilized. We start from a small outlier-free subset robustly chosen as the major region, where an octree is used for accelerating computation. Then the region is iteratively increased by adding samples at a time. Finally, by treating the points on minor regions as outliers, we are able to define the principal axes of the shape as one of the major region. One of the advantages of our algorithm is that it automatically disregards outliers and distinguishes the shape as the major and minor regions during the principal axes determination without any extra segmentation procedure. The presented algorithm is simple and effective and gives good results for point-based shapes. The application on shape alignment is considered for demonstration purpose.

Keywords

Principal axes; Point-based shapes; Least median of squares (LMS); Forward search; Principal component analysis (PCA)

1 Introduction

One of the important tasks in computer graphics and computer vision is the determination of *location* and *orientation* of an object within a specified frame of reference [1,2]. Typically, this information is also called the *pose* of the object. There are several representations of an object's pose, including the principal axes, affine transformation, moment invariants, medial axis transform, and others [1]. The simplest and most widely accepted one for 3D shapes is based on the principal axes of the object [1], which completely consists of its orientation and position with respect to an orthogonal frame or coordinate system. One main problem using

* Corresponding author. Email addresses: liuyushen00@gmail.com (Yu-Shen Liu), ramani@purdue.edu (Karthik Ramani).

traditional global techniques is that the derived principal axes might be quite different for some similar shapes [3,4].

In computer graphics and geometric modeling, 3D shapes are commonly represented by explicit surfaces, implicit surfaces, or polygonal meshes. With 3D scanners becoming the standard source for geometric data acquisition, point sets have received an increasing attention as an alternative shape representation [5–11]. The point-based representation allows more flexibility when a globally consistent surface topology is not necessarily required [8], and can cover wide shapes, such as non-manifold. A geometric shape referred to in this paper is represented by a set of points sampled from its underlying surfaces. Our method presented in this paper is also suitable for other shape representations through converting them into point sets.

In this paper, we focus on the problem of the principal axes determination for 3D shapes represented by point sets, possibly with noise, by combining robust statistics methods.

1.1 Previous work

The principal axes or pose determination of 3D shapes is one of the most important techniques in robotics, computer graphics, and computer vision. It is used in many areas, such as shape alignment [12], object recognition [13], generating 2D drawing views from 3D models [14]. There have been some effective approaches in extracting the pose of both polyhedral and smooth objects by associating some additional information, such as the surface normals and areas, except the point position. These approaches include extended Gaussian image (EGI) [13], complex extended Gaussian image (CEGI) [2], and some related improvement techniques [15,14]. A review of many available methods for both polyhedral and smooth objects is beyond the scope of this paper. The reader may consult Refs. [2,15,14] for detailed expositions. In this section, the work most related to ours will be reviewed.

Research on determining the principal axes of 3D shapes can be categorized as the global and partial approaches. The most common technique for determining the principal axes of 3D shapes is principal component analysis (PCA) that is a global approach. The main advantages of PCA are simple, fast, and applicable to most of 3D models, also including non-manifold and point sets. Each object is analyzed by PCA in three principal axes (or eigenvectors), and according to their eigenvalues these vectors are mapped to three axes. PCA is generally considered to work well for a variety of classes. The PCA approach seeks a projection that best represents the data in a least squares sense [16]. One problem for the least squares method is lack of robustness. Indeed, one single outlier can have an arbitrarily large effect [17]. The principal axes derived by PCA might be quite different for some similar shapes due to some small local difference between shapes [3]. Our main goal in this paper is to ignore the effect on the minor regions of 3D shapes during the principal axes computation.

More recently, Passalis et al. [12] introduced a method for improving PCA by combining the symmetry planes of a 3D model. Their method relies on the assumption that most of real life objects are symmetrical with respect to a plane. First, a symmetry plane of the model is determined using a global optimization technique. Then the perpendicular vector to the symmetry plane is used as the first axis, and the remaining two axes are determined by projecting all vertices of the model onto the plane of symmetry and performing a 2D PCA. This strategy actually reduces 3D into 2D by projecting 3D points into the symmetry plane, but for the 2D case there is still the same problem as 3D using PCA. Certainly, for classes of objects with more symmetry planes, their method can completely eliminate PCA by seeking the primary and secondary symmetry planes. However, most 3D life objects contain deformation, so multiple symmetry planes are difficult to be found. In addition, the determination of symmetry is also a non-trivial task [18,19].

In general, the partial approaches first segment the model into some patches, and then the patches on major regions are used for the final pose computation. Recently, Gal and Cohen-Or [4] presented a partial shape matching method for shape alignment. Their method first segments the models into some salient geometric features, and then searches for matching between pairs of salient features from each model. This partial method can produce better results than those using PCA applied to the whole model. However, segmentation is a non-trivial task itself. It often fails to segment surfaces sampled from non-manifold or large and complex point sets acquired by 3D scanning devices, and consequently results in an unsuccessful pose determination. Furthermore, both segmentation and partial matching require the expenditure of large amounts of time and space if the number of points and patches is gigantic.

Another class of approaches focuses on topological or graph comparison for determining the pose. The topology-based approaches rely on the fact that 3D model topology is an important shape characteristic. For instance, Hilaga et al. [20] proposed the use of the skeleton of an object for retrieving. A drawback of topological approaches is that relatively small anomalies on the object's surface can have a significant impact on the topological properties of the object [12]. In addition, the topological approaches also have some problems similar to partial shape matching methods for non-manifold and complex objects.

1.2 Robust statistics methods

Determining the principal axes of point sets using PCA is similar to the linear regression using least squares [16]. The linear regression belonging to a statistical method is considered to be *robust* if it has a large *breakdown* point. A breakdown point might be loosely defined as the smallest percentage of outliers that can cause the estimator to take an arbitrarily large aberrant values [17,7]. For instance, the breakdown point of the median of a set of values is 50% [17], whereas least squares has a breakdown point of 0%.

Robust statistics methods have been applied to various computer vision applications [21]. For example, Torre and Black [22] proposed a robust PCA approach for automatic learning of linear models from data that may be contaminated by outliers. However, there is little attention on 3D computer graphics. Jones et al. [23] and Fleishman et al. [24] have applied the bilateral filter to mesh denoising, which can be considered as a robust statistics technique. Pauly et al. [25] introduced a method for analyzing the uncertainty and variability of a point set. Their method can be regarded as a *backward* method that can not detect *masked outliers* [7]. Masked outliers are outliers that can not be identified from the statistics of a model that is dealt with to the entire sample set. The strategy of backward methods for fitting a model first fits a model to the entire sample set and then tries to delete bad samples. The reader may consult Ref. [7] for other several works related to robust statistics methods in computer graphics.

Fleishman et al. [7] recently presented a new robust fitting method from a set of points in order to overcome the drawback of backward methods. The main tool that they use is the *forward* search algorithm which has a significant advantage in detecting outliers over commonly used backward methods. The main strategy of their algorithm is as follows. A subset of the data is first fitted using the second degree polynomials based on the forward search algorithm, and then the rest of the data is identified as outliers. To fit multiple surfaces, the above procedure is repeated for the remaining point sets. Our work presented in this paper is in the same spirit and applies the forward search to the principal axes determination.

1.3 Contributions

The work most related to ours is Fleishman et al.'s work [7]. From statistical view point, the method in [7] treats the points across the discontinuities as outliers in order to define sharp

features. Unlike their method for defining a surface from a point cloud, our goal is to determine the principal axes of 3D shapes. Our work is based on a powerful statistic technique, called least median of squares (LMS) [17,26], to improve the PCA limitations. Using this method, an outlier-free major region of a 3D shape is extracted, which ignores the effect on other minor regions regarded as the outliers of the shape. At the last phase of our algorithm, the principal axes of the major region are regarded as the final ones of the shape. In order to effectively approximate the LMS optimization, the forward search technique [27] is utilized. The basic idea in forward search is to start from a small subset of robustly chosen samples of the data that excludes outliers. To accelerate the extracting of the initial subset, we exploit the octree for approximating the shape and sampling points. Then the principal axes are computed iteratively by adding one sample into the subset at a time while monitoring certain statistical estimates. We can use the method to deal with noise, outliers, and minor regions on shapes. The presented algorithm is simple and effective and can give good results for point-based 3D shapes. The application on shape alignment is considered for demonstration purpose. The main contributions of our work can be summarized as follows:

- Propose a new robust technique for principal axes determination by guiding the classical PCA computation based on least median of squares and the forward search technique. The proposed algorithm automatically disregards outliers and distinguishes the shape as the major and minor regions during the principal axes determination. Our algorithm can be applied to large and complex point sets acquired by 3D scanning devices or sampled from multi-surfaces (or non-manifold).

- The octree-based approximation and point sampling methods accelerate the extraction of the initial subset in forward search.

- Apply our algorithm to some shape matching techniques, such as shape alignment.

- Investigate the effect on noise and sampling density and compare our method with previous works.

2 Robust estimation

In this paper, we consider the following input conditions. For this case in 3-dimensional space, let $\mathcal{C}_N = \{\mathbf{p}_i | i = 1, \ldots, N\}$ be a set of unorganized data points, where $\mathbf{p}_i = (x_i, y_i, z_i)^T$ is a 3D vector. The set $\mathcal{C}_N$ of data points is assumed to be a sampling of underlying surfaces of a 3D shape with or without boundary. We suppose that the unorganized data points, often referred to a *point set*, *point clouds*, or *scattered data points* in the literature, may have non-uniform distribution with considerable noise.

The most commonly used technique for determining the principal axes of 3D shapes is the PCA technique. In this section, we will investigate the reasons and limitations on the application. Then a robust statistics method, i.e. least median of squares (LMS), will be introduced to overcome the PCA limitations. Finally, the forward search technique is utilized in order to effectively approximate the LMS optimization.

2.1 Review of PCA

In statistics, PCA is a technique for simplifying a data set by reducing multidimensional data sets to lower dimensions for analysis. The basic idea of PCA is to seek a projection that best represents the data in a least squares sense [16]. We first review this procedure by considering a 3D shape represented by a point set $\mathcal{C}_N$. The specified reference frame for the 3D shape consists of an origin and three principal axes. PCA assumes the origin is at the sample mean point **o**. PCA wants to find a vector **e** through the origin **o** such that the sum of the squared distances between various $\mathbf{p}_i \in \mathcal{C}_N$ and the corresponding projection point $\mathbf{p}_i^*$ onto **e** is as small as possible. The squared-error criterion function is defined by

$$J(\mathbf{p}_1^*, \ldots, \mathbf{p}_N^*, \mathbf{e}) = \sum_{i=1}^{N} \|\mathbf{p}_i^* - \mathbf{p}_i\|^2 . \quad (1)$$

Let **e** be a unit vector. Then the equation of the line through the origin **o** in the direction of **e** can be written as $\mathbf{x} = \mathbf{o} + \alpha\mathbf{e}$, where the scalar α corresponds to the distance of any **x** from the mean **o**. If each projection point $\mathbf{p}_i^*$ of $\mathbf{p}_i \in \mathcal{C}_N$ onto the line is represented by

$$\mathbf{P}_i^* = \mathbf{p}_i^*(\alpha_i) = \mathbf{o} + \alpha_i \mathbf{e}, \quad \alpha_i \in \mathbb{R}. \quad (2)$$

We can find an optimal set of coefficients α_i by substituting Eq. (2) into Eq. (1) and minimizing the squared-error criterion function

$$J(\alpha_1, \ldots, \alpha_N, \mathbf{e}) = \sum_{i=1}^{N} \|(\mathbf{o} + \alpha_i \mathbf{e}) - \mathbf{p}_i\|^2. \quad (3)$$

By adding the constrain $\|\mathbf{e}\| = 1$ and setting the derivative to zero for finding the best direction **e**, the solution to this problem involves the so-called *covariance matrix* **A** [16] defined by

$$\mathbf{A} = \sum_{i=1}^{N} (\mathbf{p}_i - \mathbf{o})(\mathbf{p}_i - \mathbf{o})^T. \quad (4)$$

The eigenvector corresponding to the largest eigenvalue of the covariance matrix **A** is the first principal axis **e**.

In fact, PCA is similar to the linear regression in a least-squares sense. However, a single sample with a large error, an *outlier*, can change the principal axes arbitrarily. This results in that the derived principal axes might be quite different for some similar shapes [3,4].

2.2 Least median of squares

To overcome the lack of robustness using least squares in Eq. (1), some robust methods might be used for improving PCA, such as making use of some weight functions for bounding the influence of outliers. However, most robust methods are *least sum of squares* by replacing the square by something else, and they can not raise a high breakdown point [17].

In our case, we assume that a 3D shape represented by a point set $\mathcal{C}_N$ consists of two parts: a *major* region and the remaining *minor* regions, and there is no overlap between them. The major region is expected to contain at least 50% points of the entire point set, so the remaining minor regions have up to 50% points. In our work, not only the noise but also the minor regions are considered as outliers for determining the principal axes of the point-based shape. Our motivation is to improve the least sum of squares in PCA with a high breakdown point (up to 50%). This above assumption, i.e. the major and minor regions making up of the 3D shape, is similar to the partial shape alignment [4], in which the major region is defined by the set of some salient features and the remaining can be considered as minor regions.

The *least median of squares* (LMS) is a robust regression method that estimates the parameters of the model by minimizing the median of the absolute *residuals*. In other words, LMS replaces the sum of least squares by a median. LMS satisfies a 50% breakdown point [17]. The resulting estimator using LMS can resist the effect of nearly 50% of contamination in the input data, which is applicable to our case. In our case, we define the absolute residual as the distance between the test point $\mathbf{p}_i$ and the projection point $\mathbf{p}_i^*$ onto the first principal axis through the origin: for the ith point $r_i=\|\mathbf{p}_i^* - \mathbf{p}_i\|=\|(\mathbf{o}+\alpha_i\mathbf{e}) - \mathbf{p}_i\|$. In this paper, we search a best direction $\mathbf{e}$ that minimizes the median of the residuals as follows:

$$\min_{\mathbf{e}} \operatorname*{median}_{i} \|(\mathbf{o}+\alpha_i\mathbf{e}) - \mathbf{p}_i\|, \tag{5}$$

where $\mathbf{e}$ is the first principal axis that will be computed. Rousseeuw [17] has also pointed out there always exists a solution for LMS.

Eq. (5) can be solved using the following random sampling algorithm (i.e. RANSAC) [7,28]. First, k points are selected at random, and the first principal axis is computed using the standard PCA algorithm to the points. Next the median of the residuals of the remaining $N-k$ points is computed. The process is repeated T times to generate T candidate axes. The axis with the minimal median is selected as the final principal axis $\mathbf{e}$. For the remaining two principal axes, we might use the similar strategy acquired by projecting all points onto the plane perpendicular to $\mathbf{e}$ and through $\mathbf{o}$ and performing a 2D resolution for Eq. (5). A small value of k does not use all of the available points to PCA computation, while a larger value of k requires more iterations. If k is too large, the algorithm becomes sensitive to outliers including noise and minor regions.

2.3 The forward search algorithm

The forward search algorithm [27] is a robust method that avoids the need to fix k. Fleishman et al. [7] applied this technique to reconstruct surfaces from point clouds. The forward algorithm first searches a small outlier-free subset and then iteratively refines the subset by adding one sample at a time. This is in contrast to the backward algorithms, which first deal with the entire data points and then delete bad samples. Fleishman et al. [7] show that some outliers with a large breakdown point usually fail on fitting based on the backward algorithms, whereas the forward algorithm will give satisfactory results. The initial model is computed for Eq. (5) using the LMS method with a small k value, typically k close to p for a model with p parameters [27,7] (p = 3 in the 3D case). In our implementation, we choose k = 4.

During the forward search, a number of parameters can be monitored to detect the influential points. Typically, the forward search will add the good-samples first and only when these are exhausted, outliers will be added. Atkinson et al. [27] suggested several statistics, including the residual-plot, Cook's distance and others, to be monitored. For their purposes, these are plotted on a graph and inspected visually. The maximal residual $r_{\max}$ is monitored by Fleishman et al. [7]. In our technique, we also monitor the maximal residual similar to Fleishman et al.'s strategy [7]. However, we compute $r_{\max}$ based on the initial subset. It will be discussed in Section 3.3.

Using the forward search technique for solving Eq. (5), we present the main procedure of determining the principal axes of a point set $\mathcal{C}_N$ as follows:

1. Choose a small outlier-free subset Q using LMS.
2. The principal axes and the origin are computed using PCA to Q.
3. The point with the lowest residual in the remaining points is added into Q.

4. Repeat steps 2 and 3 until the error is larger than a predefined threshold r_{max} and identify the points in $\mathcal{C}_N - Q$ as outliers.

Fig. 1 shows an example of this process in two dimensions (see caption). The process is simple, but there are some limitations for effective computation for large and complex point sets acquired by 3D scanning devices. We will take advantage of the octree and point sampling to significantly accelerate the process and improve its stability. The overall algorithm will be introduced in the next section.

3 Robust principal axes determination

3.1 Initial robust estimator

In the first step of the forward search algorithm, the initial subset is computed using the LMS algorithm with a small k value (see Section 2.3). If the number N of points in $\mathcal{C}_N$ is small, the choice of the initial subset can be performed by exhaustive enumeration of all $\binom{N}{k}$; otherwise, the LMS uses the RANSAC algorithm (see Section 2.2) that requires a large iteration number T to achieve a high probability of finding a good estimator. The LMS, as a statistical method, assumes that the samples (points) are independent. If g is the probability of selecting a single good sample at random from the original point set $\mathcal{C}_N$, then the probability P of successfully finding k good samples after T iterations can be computed by $P = 1 - (1 - g^k)^T$ [7]. Furthermore, for every iteration, LMS requires a sort of the residuals of the remaining $N - k$ points to find their median, so this will also require the expenditure of large amounts of time and space for sorting for a large T if N is gigantic. In general, there is a large number of points for a 3D shape acquired by 3D scanning devices. For instance, Fig. 2 shows a 3D shape with 23,400 points, and the running time for finding an initial subset ($k = 4$) is about 401.8 seconds with $T = 5000$ iterations, where the process of computing and sorting the remaining residuals is repeated 5000 times.

3.1.1 Octree-based approximation—In this phase, we use octrees described by Adams et al. [29] to accelerate the initial subset searching, where the points of $\mathcal{C}_N$ are considered as surfels with zero radius. In some point-based processing, such as Boolean operations [29], surface reconstruction [30], and area computation of point-based models [9], the octree can be utilized. Thus, it is not an additional price for the point set. In the preprocessing step, an axis-aligned octree of depth d [29] can be constructed. Adams et al. [29] suggested that $d = 4$ or 5 can lead to both a small approximation error for shapes and little computation time. We typically choose $d = 5$ in our implementation.

Suppose that we have constructed an octree for $\mathcal{C}_N$, and have classified the cells of the octree as two types: boundary cells containing points of $\mathcal{C}_N$, and empty cells containing no point of $\mathcal{C}_N$ [29]. For each boundary cell of the octree, we compute a center of points in the boundary cell as a *feature* point that characterizes all points inside the boundary cell. Let $\mathcal{F}$ be the set of feature points for all boundary cells of the octree. For every iteration, k points are first selected at random from the original point set $\mathcal{C}_N$, and three principal axis are determined using PCA. Let **e** be the first principal axis obtained using PCA. Next the residuals are computed by projecting feature points in $\mathcal{F}$ onto **e**. Unlike the original LMS method, we compute the median of the residuals of feature points in $\mathcal{F}$ instead of ones of $N - k$ points in $\mathcal{C}_N$. The octree-based approximation yields both the good approximation results and little computation time. Some other tree data structure can also be used for approximating the point set, such as B-Trees [11].

3.1.2 Octree-based point sampling—In the first step of the forward search algorithm, k points are selected at random for every iteration for computing three principal axes. Point

sampling is an intermediate step for a variety of computer graphics applications, and specialized sampling strategies have been developed to satisfy the requirements of each problem (such as Refs. [31–33]). A simplest sampling strategy is to choose k points randomly from N points in $\mathcal{C}_N$ using pseudo-random number generators, but it might occur the chance of point clustering. For example, k points appear on the same boundary cell of the octree of $\mathcal{C}_N$, which results in an invalid sample. Plenty of invalid samples will result in a low convergence rate.

Instead, our method for generating unbiased random points with respect to the number of points in each boundary cell proceeds as follows. First, for each boundary cell, we count the number of points inside the boundary cell, which is also called the *density* of the boundary cell. Then, we store the density of each boundary cell in an array along with the cumulative density of boundary cells visited so far. Next, k boundary cells are selected with probability proportional to their densities. This procedure is finished by generating k random numbers between 0 and the total cumulative density and performing a binary search on the array of cumulative densities. For each selected boundary cell, one sample point is chosen randomly from the points inside the boundary cell. Intuitively, the above sampling method gives k uniform random points with respect to the density of the boundary cells. Our idea is similar to the area-based sampling strategy presented by Osada et al. [33]. They uniformly sample points from a triangle mesh with probability proportional to the area of triangles.

We find that the sampling based on octree can offer a faster convergence rate than one using the direct sampling from points of the original model. Of course, other sampling methods that sample according to curvature or other surface properties would be possible as well. However, these methods usually require the large expense of computation time, and this is not applicable to fast computation in our applications.

Fig. 2(a) illustrates an octree of depth $d = 5$ for a point-based shape, where the octree has 499 boundary cells that are displayed. In Fig. 2(b), a good initial subset with $k = 4$ points (red) is obtained with $T = 5000$ iterations using the above octree-based approximation and point sampling methods, and its running time is about 1.822808 seconds.

3.2 Residual computation

In Section 2.2, we defined the residual as the distance between the test point $\mathbf{p}_i$ and the corresponding projection point $\mathbf{p}_i^*$. Eq. (5) minimizing the median of projection distance onto the first principal axis is an extension of PCA. In the special 2D case, LMS corresponds to finding the narrowest “strip” covering half of the points [17]. In our case, the optimization in Eq. (5) also implies that the major region of 3D shapes is nearly a *cylinder-like* shape surrounding the first principal axis. The assumption can work well for most mechanical parts (such as pipeline and bearing) and real life objects (such as human, animal, and tree). However, there are still some limitations for point sets sampled from surface patches. The cylinder-like shape is not expected to be similar for most surface patches whose major regions are near planes. To overcome the disadvantage for surface patches using LMS, we might redefine the residual as the distance between the test point $\mathbf{p}_i$ and the projection $\mathbf{p}_i^*$ onto the reference plane defined by the first and second principal axes. Based on the new residual definition, the LMS optimization can be implemented by projecting all points onto the reference plane, which is determined by two principal axes using PCA, at every iteration. Fig. 3 shows an examples for determining the principal axes for a point set sampled from surface patches using the new residual definition. Of course, some other residuals can also be defined for being suitable for some special style of shapes. For some shapes without obvious major regions, our algorithm is also incapable as the standard PCA.

3.3 Forward search on point sets

After a robust estimator is computed for a small number of points using the above algorithm, the results are three reference principal axes and an initial outlier-free subset Q. The threshold of maximal tolerated residual $r_{\max}$ is computed using Q as follows. The k points in Q are projected onto the first principal axis (or a reference plane in Section 3.2). Suppose that r_t is the maximal residual of k points. We define $r_{\max}$ to be proportional to r_t

$$r_{\max} = \lambda r_t, \tag{6}$$

where λ is the value of residual band. We typically use λ = 1.25. A small value of λ does not use all of the available samples to determine the major region, while a larger value for λ requires more iterations. The second step of the forward search is to iteratively add one point with lowest residual to the set Q at each iteration by sorting the residuals of the remaining points. The iteration is terminated until the lowest residual is larger than $r_{\max}$.

Since we expect a less iteration and little computation time for the forward search, instead of only adding one point at each iteration, we add more points every time. The number m of adding points can be set by users. We typically choose m = 60 or more for the dense point sets. Fig. 2 illustrates the procedure of robust principal axes determination for a 3D shape.

Algorithm 1 : RobustPrincipalAxes($\mathcal{C}_N$, **o**, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, Q)

Input:

$\mathcal{C}_N \in \mathbb{R}^{3\times N}$: the given point set with N points

Output:

o: the origin of the reference frame

$\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$: the first, second, and third principal axes

$Q \subseteq \mathcal{C}_N$: the working subset, i.e. the major region

Local variables:

k: the number of random samples

$r_{\max}$: the maximal tolerated residual

I: the current iteration

$c_{\text{rem}} \subseteq \mathcal{C}_N$: the set of the remaining points

m: the number of points added into Q at every iteration

MAX_ITERS: the maximal number of iterations

begin

1: $Q \Leftarrow \emptyset$;

2: **LMS**($\mathcal{C}_N$, k, Q);

3: Compute $r_{\max}$ using Q via Eq.(6);

4: $I \Leftarrow 0$;

5: **while** (I + + < MAX_ITERS) **do**

6: Compute the origin **o** and three principal axes: $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ for Q using PCA;

7: $c_{\text{rem}} \Leftarrow \mathcal{C}_N - Q$;

8: Compute the residual as the distance between each point in c_{rem} and its projection point onto the line through **o** in the direction of $\mathbf{e}_1$;

9: Get m points with lowest residuals for c_{rem};

10: **if** (the maximal residual of the m points > $r_{\max}$) **then**

11: The points whose residuals are smaller than r_{max} are added into Q;

12: **return**

13: **end if**

14: Add the m points into Q;

15: **end while**

16: Project $\mathcal{C}_N$ onto the plane determined by $\mathbf{e}_2$, $\mathbf{e}_3$ and $\mathbf{o}$, then perform a 2D robust PCA for updating the principal axis $\mathbf{e}_2$ and $\mathbf{e}_3$;

end

3.4 The algorithm implementation

The outline of an algorithm for robust principal axes determination, called **RobustPrincipalAxes,** is given in Algorithm 1. The algorithm takes as input a point set $\mathcal{C}_N$ and computes the origin $\mathbf{o}$ and three principal axes $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ of the reference frame. This is achieved through an iterative procedure with the aid of a variable Q which is a working subset of $\mathcal{C}_N$. Initially, Q is computed using the LMS algorithm through selecting k points at random with T iterations, as illustrated in Algorithm 2. Here, the LMS procedure is accelerated for approximating and sampling using an octree of depth d for $\mathcal{C}_N$. The octree can also be constructed in the preprocessing step.

Algorithm 2 : LMS(c_n, k, Q)

Input:

$c_n \in \mathbb{R}^{3\times n}$: the working point set with n points

k: the number of random samples

Output:

$Q \subseteq c_n$: the initial subset

Local variables:

d: the depth of the octree for c_n

T: the number of iterations

$c_{temp} \subseteq c_n$: the initial subset

$\mathbf{o}$: the origin of the reference frame

$\mathbf{e}_1$: the first principal axis of the reference frame

r_{half}: the median of redsiduals

r_{min}: the minimal redsidual

begin

1: Construct the octree of depth d for c_n; /* *the octree can also be constructed in the preprocessing step* */

2: $r_{min} \Leftarrow \infty$**;**

3: **for** ($i = 0; i < T; i++$) **do**

4: Select randomly a subset c_{temp} with k points using the octree-based sampling;

5: Compute the first principal axis $\mathbf{e}_1$ and the origin $\mathbf{o}$ for c_{temp} using PCA;

6: **for** (boundary cells of the octree of c_n) **do**

7: Compute the residual as the distance between the feature point of each cell and its projection point onto the line through $\mathbf{o}$ in the direction of $\mathbf{e}_1$;

8: **end fors**

9: Compute the median r_{half} by sorting the residuals;

10: **if** ($r_{half} < r_{min}$) **then**

11: $r_{min} \Leftarrow r_{half}$;

12: $Q \Leftarrow c_{temp}$;

13: **end if**

14: **end for**

end

According to Algorithm 2, the proposed algorithm is passed into a point set ($c_n \Leftarrow \mathcal{C}_N$), and a loop with T iterations begins. At each iteration, a subset c_{temp} with k points is randomly selected using the octree-based sampling, and the first principal axe $\mathbf{e}_1$ and the origin $\mathbf{o}$ is computed for c_{temp} using the classical PCA. Then, the residuals of c_n are calculated as the distance between the feature point of each cell and its projection point onto the line through $\mathbf{o}$ and $\mathbf{e}_1$. Next, the median r_{half} of the residuals is obtained. If r_{half} is less than the minimal residual r_{min}, r_{min} and Q are updated as r_{half} and c_{temp}, respectively.

During the iterative procedure in Algorithm 1, the cardinality of Q is gradually increased by adding m points with lowest residuals every time. In this way, one is able to increase Q regarded as the major region in the forward search. If the residuals of the remaining points ($\mathcal{C}_N - \mathrm{Q}$) are more than a threshold r_{max}, the procedure is terminated. Finally, the points in the major region Q are used to compute the first principal axis $\mathbf{e}_1$, and the remaining points are identified as outliers or minor regions.

Algorithm 1 is mainly used to compute the first principal axis $\mathbf{e}_1$. The remaining two principal axes might be computed using the strategy similar to Ref. [12] by projecting all points of $\mathcal{C}_N$ onto the plane perpendicular to $\mathbf{e}_1$ and through the origin $\mathbf{o}$. Then a 2D case for Algorithm 1 is performed for computing $\mathbf{e}_2$ and $\mathbf{e}_3$.

4 Results and applications

We have implemented the technique presented in the previous section and tested it on a large number of different point-based 3D shapes. The algorithm described above is implemented in C++. In this paper, the execution time is given in seconds on a Pentium IV 1.70GHz processor with 512M RAM excluding the time of loading point sets.

Before computing the principal axes, a preprocessing step includes an octree construction that can be performed in a short time. Table 1 gives the time in seconds for some point-based shapes referred to in this paper, where "*N*" is the number of points of the models, "Major%" is the percentage of the major region that belongs to the original point set, "*m*" is the number of points added at each iteration (see Section 3.3), "T1" and "T2" are the time of constructing an octree and computing the principal axes. The computation time increases with the number of points of the models used. The time also depends on the size of major region and the number of iterations. In all examples, we use $T = 5000$ iterations in Algorithm 2.

4.1 Testing major regions for articulated and scanned models

One important issue of principal axes computation using our method is its robustness of computing the major regions for different poses of the same object. We test the 3D models with multiple poses in Gal et al.'s database [34], which contains some similar articulated models with multiple poses. All models in this database are represented by triangled meshes, so we first sample the triangled meshes to generate the point sets as the tested models. Then our method is applied to the point-based models for obtaining the major regions. Most of real life objects in this database, such as animal models, contain the similar major region with a cylinder-like shape. Fig. 4 shows some results of major regions for the dinosaur and horse models, respectively. Here, our method can recognize that the body of animal is the major region. The major regions computed by our method are insensitive to pose changes of the same object.

In addition, some real point sets acquired by 3D scanning devices are also tested for computing the major regions and the final principal axes (see Fig. 5). These selected models are obtained from AIM@SHAPE Shape Repository. Fig. 5(a) shows the result of a dancer model. In this example, the computed major region, which contains automatically the portions: one leg, the head and hands, and part of body, captures the main pose of the dancer. In Fig. 5(b), another complex example is given for the Neptune model. Although the lance and pedestal hold part of the model, the principal axes on the major region are still determined on the Neptune's body. In Fig. 5(c), the major region is captured as the body of the pig, not its legs. Our results show that our method works good on both scanned or articulated objects.

4.2 Shape alignment

Many shape matching methods require an alignment step in order to position all objects in a standard orientation [4,12]. These searching methods rely on the ability to align models by some global similarity transformation (rotation + uniform scale) that normalizes the models and establishes some correspondence between them. Most global alignment methods first determine a rigid-body transformation and a uniform scale, which align two models together as closely as possible, before measuring the distance between them. This is typically achieved by PCA applied to the whole model. The PCA method does not discriminate between the major regions, and can easily cause similar local features to be misaligned. Our alignment algorithm based on robust principal axes determination first searches the major region of each model. Then for each model, the origin and three principal axes are computed for the corresponding major region as a specified frame of reference. Finally, the transformation is applied to two reference frames for finishing the final alignment.

In Fig. 6, a 3D example is shown for aligning two gun models, where the gun in Fig. 6(b) is part of one in Fig. 6(a) only through removing the cartridge clip. Fig. 6(c) shows that a global PCA alignment fails to align them correctly, whereas our method correctly aligns as shown in Fig. 6(d). Comparison between two major regions for Figs. 6(a) and 6(b) is given in Fig. 7. Another example is shown in Fig. 8, where two point sets are sampled from the drills with different angle-rotation hands. Our alignment (see Fig. 8(d)) is better than a global PCA alignment (see Fig. 8(c)).

In addition, an animal example is shown in Fig. 9, where the same models appear in two different poses. In spite of the obvious difference between them, our algorithm aligns the two better than a global PCA alignment, where the body of the cat is automatically identified as the major region not its legs. Our alignment is based on a majority scheme that finds the transformation which satisfies the major regions not the whole shapes.

5 Discussion

In this section, we will discuss some parameters used in our algorithm, influences on noise and irregular samples, comparison with previous works, and limitations of our algorithm.

5.1 Parameters

In Algorithms 1 and 2, the parameters of the algorithms are: $r_{\max}$, T, m, and MAX_ITERS. In our implementation, the maximal tolerated residual $r_{\max}$ in Eq. (6) is determined by both the scale λ and the initial outlier-free subset Q, as described in Section 3.3. Eq. (6) can be regarded as a rough approximation for the narrowest strip covering half of the points in $\mathcal{C}_N$. The reason of choosing Eq. (6) as the threshold during monitoring is that the principal axes of Q approximates the final principal axes *of* C_N. In general, when the good samples are added in the forward search, the updated principal axes will change a little; there will be a clear change, otherwise. One may choose a large λ and increase the number of iterations. If λ is too large so

that the major region is equal to $\mathcal{C}_N$, the principal axes using our method is same as ones using PCA. In some sense, PCA is only one special case of our algorithm.

The maximal residual—Fig. 10 shows the angle difference between the initial and final first principal axes **e** with respect to the various scale λ of $r_{\max}$ in Eq. (6) for two models, which are referred to in Table 1. We vary the threshold of the maximal residual, using λ from 0 to 2.5, to determine its effect on the the initial and final first principal axes. The value of angle difference between them increases with λ until the major region is equal to $\mathcal{C}_N$. This reason is that the forward method adds the points with the minimal residual into the current subset at each iteration until all point sets are exhausted. We observe that the value of λ between 1.0 and 1.5 can leads to a small angle change between the initial and final first axes. For example, λ = 1.25 can get an angle difference close to 2 degree. In fact, we found the initial subset can introduce the good direction for the first principal axis, but it is not sufficient for the center computation due to the few points in the subset, as described in Fig. 1. The main goal of the forward searching procedure in our method is to mainly refine the center position and slight adjust the direction for the frame of reference.

Eq. (6) is only a simply linear function for estimating the bound of the major region. A potential improvement considered is to choose a non-linear function instead of Eq. (6), such as Gaussian function. In fact, some standard methods in regression analysis for identifying outliers, such as the *residual plot* [27], can also be used for monitoring the termination of iterations during the forward search. In the future we plan to take advantage of the statistics methods to find a way to automatically select $r_{\max}$ such that the forward search is adaptively achieved.

The forward search algorithm as described in Section 2.3 adds a single sample into the current subset at each iteration. In our implementation that presented in Section 3.4, we allow adding multiple (m) points at each iteration as long as their residuals are within the allowed tolerance. For sparse point sets, a small m can be considered. Furthermore, in all results shown in this paper, we use $T = 5000$ iterations in Algorithm 2 for obtaining both small errors and little computation time. The maximal iteration number MAX_ITERS is usually a predefined integer to ensure that the number of points on the major region is more than 50% in $\mathcal{C}_N$ (generally $[0.5N/m] \leq \text{MAX\_ITERS} \leq [N/m]$).

Multiple objects—The method presented in this paper focuses on a set of points with no connectivity. This permits one to deal with multiple and non-manifold objects. In Fig. 12(d), we show an example from Ref. [35] for determining the principal axes for a point set sampled from two objects, where the large object is identified as the major region.

5.2 Noise and sampling density

Models created from 3D scanners usually contain noise [6,23,24]. Noise tends to increase point-error (or “point cloud thickness”) [10]. Our method uses tools from robust statistics to operate well in the presence of noise, identifies outliers and ignore them. The main tool that we use is the forward-search algorithm which has a significant advantage in detecting outliers over commonly used “backward” methods. To test the ability of the procedure to handle noise, we have added uniformly distributed random noise (along the normals with 30.0% variances of the diagonal of the bounding box of the model) to the gun model. Fig. 11(a) and 11(a) shows the original and noisy models with the major regions colored by blue. Fig. 11(c) gives the alignment result between the original model and the noisy one using our method. Our method can keep the almost consistent principal axes between two models.

Highly irregularly sampled point sets are uncommon in scanned data sets [24]. If the sampled points are not uniformly distributed over the underlying surfaces of 3D shapes, our method may lead to large errors for approximating the principal axes. The forward search method

approximates the LMS optimization using the PCA computation for the currently working point sets on each iteration, as seen in Algorithms 1 and 2. The only information used in our current implementation is the position of the points in $\mathcal{C}_N$. However, the PCA analysis is blind to the position of the points [11], even though we only use PCA for computing a subset of $\mathcal{C}_N$. Recently, Kalaiah et al. [11] presented a randomized rendering method for a point-based model. They use multi-attribute PCA to represent the geometry and attributes of the point-based model in a high dimensional space. In their case, the input is a set of N points with three attributes: spatial position, normal, and color. The mean, variance, and the basis of each of these attributes are identified respectively to obtain the high dimensional eigenvectors. In practice, normal and color of points can be estimated from scanners. To overcome in part the effect on irregular data sets, we might also apply the additional attributes of the points, including normal and color, to achieve the intermediate PCA computation in Algorithms 1 and 2. In addition, another hopeful strategy is to pre-compute the sampling densities of points as weights of points, and achieve a weighted PCA in Algorithm 1.

5.3 Combination with ICP

One main application of our method is shape alignment. Another widely used geometric alignment technique is Iterative Closest Point (ICP) [36,37], which is employed to match two clouds of points to reconstruct 3D surfaces from different scans, to localize robots, to match bone models with measures in realtime, etc. The ICP algorithm starts with two point clouds and an initial guess for their relative rigid-body transform, and iteratively refines the transform by repeatedly generating pairs of corresponding points on the models and minimizing an error metric. One main limitation of ICP and its variants is that, as a local optimization method, it is not guaranteed to find the globally optimal alignment. Therefore, ICP is only effective when the initial position of the input shapes is close to the correct alignment [36,37]. For the shape registration application, PCA is typically used to compute an initial guess between two input models, and then the initial guess is refined with ICP for finding the final transform. Our method can be expected to obtain a better initial position than PCA for improving the robustness of the ICP step.

5.4 Comparison with previous works

Backward vs. forward methods—Although there are different types of robust statistical methods (such as SVD [38], Least Trimmed Squares (LTS) [38], Iterative Reweighted Least-Squares (IRLS) [39], and RANSAC [28] techniques) being available for improving PCA computation that are used in solving practical problems in computer vision, most existing works are backward methods and they do not work well in our case. Backward methods fit a model to noisy data work by fitting a model to the entire sample set and then trying to delete bad samples. Backward methods identify the outliers with respect to the initial guess. For example, LTS first fits the data using ordinary least squares; then identifies some points with the largest residuals and discards these; finally re-fits the remaining data. One main problem of backward methods is that some large outliers may affect the final fit or principal axes computation. The main tool that we use is the forward-search algorithm which has a significant advantage in detecting outliers over commonly used backward methods [7].

Recently, Cortadellas et al. [39] presented an approaches for normalizing silhouettes of 2D images. The key technique used in this paper involves the computation of the center of gravity and the orientation of the principal axis for 2D shapes with deformation. The main idea is to improve IRLS for principal axis computation with a 2D shape dependent weighting function. This algorithm works well when being applied to 2D shapes whose measured orientation changes under slight deformations. However, it does not work well when being directly applied to our works due to two reasons. The first one is that their weighting function is not suitable for 3D point-based shapes. The presented shape dependent weighting function needs to

compute distance maps. Although, this method can be implemented for 2D images, but it is not a trivial task to determine distance maps of 3D point-based shapes without any connectivity and parameterization information. The second one is that Cortadellas et al.'s method is a backward method, which identifies the outliers with respect to the initial guess. In our experiment, we test Cortadellas et al.'s IRLS algorithm for 3D point-based shapes with the standard weight function. Since Cortadellas et al. only presented the first principal axis computation, we compare the results of four methods (PCA, IRLS, LTS and our method) for the first principal axis. The principal axis determined by IRLS perhaps might be affected by large outliers or deformation regions. Fig. 12 shows comparison of four methods for a point set sampled from two objects. PCA, IRLS and LTS are affected by the small object, while our method ignores the small object regarded as outliers and identifies the large object as the major region for the final principal axis computation and thus produces the expected result.

RANSAC vs. Forward Search—RANSAC (RANdom SAmple Consensus) is a general procedure for fitting models to data that has clearly separated outliers. Given a fitting problem with parameters **x**, the RANSAC algorithm can be described as follows [28,38].

1. Randomly select *k* data items;
2. Estimate the parameter **x**;
3. Find how many data items (of $\mathcal{C}_N$) fit the model with parameter vector **x** within a user given tolerance. Call this *Q*.
4. if *Q* is big enough, accept fit and exit with success;
5. Repeat steps 1–4 until the best fitting to the remaining data is retained.

Least Median of Squares (LMS) can be solved by RANSAC in which the median error is used to evaluate RANSAC. One limit of RANSAC for solving LMS is that it is difficult to choose an appropriate *k* value [7]. A small value of *k* does not use all of the available points to PCA computation, while a larger value of *k* requires more iterations. If *k* is too large, the algorithm becomes sensitive to outliers including noise and minor regions.

The forward search algorithm [27] is a robust method that avoids the need to fix *k*. In this procedure, a small subset of inlying points is first identified (e.g. using RANSAC or perhaps manually), and then this set is grown by iterating the following steps:

1. Add the data point with the lowest residual to the currently fit model;
2. Re-fit the model to the new set of points.

The iteration is terminated when the lowest residual is larger than some threshold reflecting an outlier. In our works, we use RANSAC for the extraction of the initial subset in forward search. Meantime, we present the octree-based approximation and point sampling for accelerating this RANSAC procedure. RANSAC is only one step of our works.

Robust least squares techniques are receiving more attention in computer graphics. For example, Pighin and Lewis [38] presented a ACM SIGGRAPH 2007 courses for an overview of the least squares technique and its robust variants for computer graphics. Using robust least squares for resolving the deformation shapes is a potential research direction by regarding deformation region as outliers.

Limitations—The first principal axis of a shape can also be used for determining shape orientation [39,40], which provides a properly oriented frame of reference and has been shown to affect performance of object recognition in the human visual system. However, note that there are many situations in which, even without noise, the principal axis approach is ill defined,

resulting in orientation estimation failing [40]. To overcome this problem, a new shape descriptor, called *shape orientability*, is introduced [40], which describes the degree to which a shape has distinct (but not necessarily unique) orientation. Orientability quantifies the likely reliability and stability of orientation estimates. For instance, even minor changes in a shape due to digitization or noise effects can substantially alter orientation estimates for shapes with low orientability. A future work is to combine shape orientability with our robust algorithm for principal axis computation of 3D point-based shape.

6 Conclusion

We have presented a robust method for determining the principal axes of point-based 3D shapes. The method is based on least median of squares (LMS) for guiding the classical principal component analysis (PCA) computation. Using the method, we can automatically identify portions of a shape as the major region or minor regions. The forward search technique is used for approximating the LMS optimization by combining the octree-based approximation and sampling. Our experiments show that the proposed method can efficiently obtain the reasonable principal axes without requiring any extra segmentation procedure. We have presented one application on shape alignment for demonstrating the effectiveness of our method. The method presented in this paper can help many point-based processing applications, such as shape registration and matching. In summary, the theoretical/methodological contributions of the paper are three-fold:

- The paper shows that interpreting PCA as a least-squares minimization drives the development of a LMS algorithm with higher breakdown point.

- The paper develops a forward search algorithm for efficiently determining/growing the major region defining the pose.

- The paper presents an efficient method for finding and initial estimator (voxel/octree-based approximation) for the forward search.

Two future works can be considered for extending the current work.

- In our experiments, we only use the point position for guiding the robust principal axes computation. The principal axes might be effected by sample density of 3D shapes. In some applications, the normals of points are also the important information for 3D shapes. Improving the robustness of this method while using sample density and normals is a topic for future work.

- Furthermore, since numerous shape-matching algorithms depend on pose estimation, we also plan to apply the robust principal axes determination to improves retrieval of point-based models by combining the known shape-matching algorithms.

Acknowledgments

We would like to thank Min Liu and Ramanathan Muthuganapathy for some helpful comments during our work. The models used in this paper were provided by the AIM@SHAPE project and Ran Gal. This material is partly based upon work supported by the National Science Foundation under Grant IIS No. 0535156. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also acknowledge partial support from the National Institute of Health (GM-075004).

References

1. Ezquerra N, Mullick R. An approach to 3D pose determination. ACM Transactions on Graphics 1996;15(2):99–120.

2. Kang SB, Ikeuchi K. The Complex EGI: A new representation for 3D pose determination. IEEE Transactions on Pattern Analysis and Machine Intelligence 1993;15(7):707–721.
3. Funkhouser T, Min P, Kazhdan M, Chen J, Halderman A, Dobkin D, Jacobs D. A search engine for 3D models. ACM Transactions on Graphics 2003;22(1):83–105.
4. Gal R, Cohen-Or D. Salient geometric features for partial shape matching and similarity. ACM Transactions on Graphics 2006;25(1):130–150.
5. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT. Point set surfaces. Proceedings of the conference on Visualization'01 2001:21–28.
6. Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT. Computing and rendering point set surfaces. IEEE Transactions on Visualization and Computer Graphics 2003;9(1):3–15.
7. Fleishman S, Cohen-Or D, Silva CT. Robust moving least-squares fitting with sharp features. Proceedings of SIGGRAPH'05 2005:544–552.
8. Kobbelt L, Botsch M. A survey of point-based techniques in computer graphics. Computers and Graphics 2004;28(6):801–814.
9. Liu YS, Yong JH, Zhang H, Yan DM, Sun JG. A quasi-Monte Carlo method for computing areas of point-sampled surfaces. Computer-Aided Design 2006;38(1):55–68.
10. Liu YS, Paul JC, Yong JH, Yu PQ, Zhang H, Sun JG, Ramani K. Automatic least-squares projection of points onto point clouds with applications in reverse engineering. Computer-Aided Design 2006;38 (12):1251–1263.
11. Kalaiah A, Varshney A. Statistical geometry representation for efficient transmission and rendering. ACM Transactions on Graphics 2005;24(2):348–373.
12. Passalis G, Theoharis T, Kakadiaris IA. PTK: A novel depth buffer-based shape descriptor for three-dimensional object retrieval. The Visual Computer 2007;23(1):5–14.
13. Horn BKP. Extended Gaussian images. Proceedings of IEEE 1984;72:1671–1686.
14. Pu J, Ramani K. A 3D model retrieval method using 2D freehand sketches. Lecture Notes in Computer Science 2005;3515:343–347.
15. Park SY, Subbarao M. Pose estimation and integration for complete 3D model reconstruction. IEEE Workshop on Application of Computer Vision (WACV2002) 2002:143–147.
16. Duda, RO.; Hart, PE.; Stork, DG. Pattern classification. 2nd. John Wiley & Sons; 2001.
17. Rousseeuw PJ. Least median of squares regression. Journal of the American Statistical Association 1984;79(388):871–880.
18. Mitra NJ, Guibas LJ, Pauly M. Partial and approximate symmetry detection for 3D geometry. ACM Transactions on Graphics 2006;25(3):560–568.
19. Podolak J, Shilane P, Golovinskiy A, Rusinkiewicz S, Funkhouser T. A planar-reflective symmetry transform for 3D shapes. ACM Transactions on Graphics 2006;25(3):549–559.
20. Hilaga M, Shinagawa Y, Kohmura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3D shapes. Proceedings of SIGGRAPH'01 2001:203–212.
21. Stewart CV. Robust parameter estimation in computer vision. SIAM Review 1999;41(3):513–537.
22. De la Torre F, Black MJ. Robust principal component analysis for computer vision, in: International Conference on Computer Vision (ICCV). Vancouver, Canada 2001:362–369.
23. Jones T, Durand F, Desbrun M. Non-iterative, feature-preserving mesh smoothing. Proceedings of SIGGRAPH'03 2003:943–949.
24. Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. Proceedings of SIGGRAPH'03 2003:950–953.
25. Pauly M, Mitra NJ, Guibas LJ. Uncertainty and variability in point cloud surface data. Symposium on Point-Based Graphics'04 2004:77–84.
26. Rousseeuw, PJ.; Leroy, AM. Robust regression and outlier detection. John Wiley & Sons; 1987.
27. Atkinson, A.; Riani, M. Robust diagnostic regression analysis. Springer; 2000.
28. Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 1981;24(6):381–395.
29. Adams B, Dutré P. Interactive boolean operations on surfel-bounded solids. Proceedings of SIGGRAPH'03 2003:651–656.

30. Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel HP. Multi-level partition of unity implicits. Proceedings of SIGGRAPH'03 2003:463–470.
31. Funkhouser T, Shilane P. Partial matching of 3D shapes with priority-driven search. Symposium on Geometry Processing 2006:131–142.
32. Nehab D, Shilane P. Stratified point sampling of 3D models. Eurographics Symposium on Point-Based Graphics 2004:49–56.
33. Osada R, Funkhouser T, Chazelle B, Dobkin D. Shape distributions. ACM Transactions on Graphics 2002;21(4):807–832.
34. Gal R, Shamir A, Cohen-Or D. Pose-oblivious shape signature. IEEE Transactions on Visualization and Computer Graphics 2007;13(2):261–271. [PubMed: 17218743]
35. Amenta N, Bern M, Kamvysselis M. A new Voronoi-based surface reconstruction algorithm. Proceedings of SIGGRAPH'98 1998:415–421.
36. Gelfand N, Mitra N, Guibas L, Helmut P. Robust global registration. Proceedings of Eurographics Symposium on Geometry Processing 2005:197–206.
37. Mitra N, Gelfand N, Pottmann H, Guibas L. Registration of point cloud data from a geometric optimization perspective. Proceedings of Eurographics Symposium on Geometry Processing 2004:23–32.
38. Pighin F, Lewis JP. Practical least-squares for computer graphics. ACM SIGGRAPH'07 courses 2007:1–57.
39. Cortadellas J, Amat J, de la Torre F. Robust normalization of silhouettes for recognition applications. Pattern Recognition Letters 2004;25:591–601.
40. Žunić J, Rosin P, Kopanja L. On the orientability of shapes. IEEE Transactions on Image Processing 2006;15(11):3478–3487. [PubMed: 17076406]

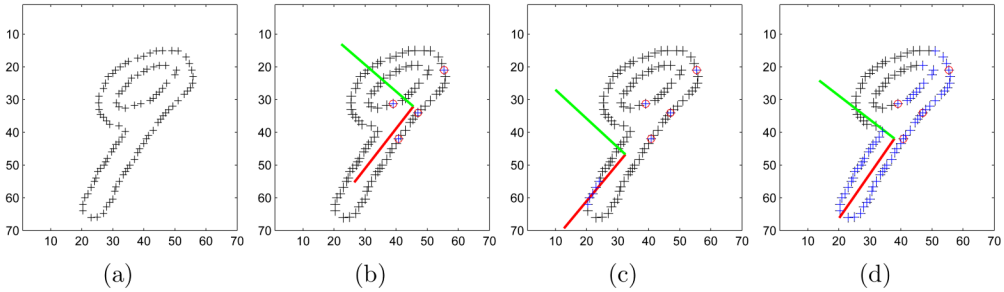

Fig. 1.

The illustration of determining the principal axes with the forward search technique. (a) The input data points sampled from the contour of a handwritten digit “9”. (b) First, determine robustly the principal axes to a small subset (4 red points) using PCA, where the red and green lines are the first and second principal axes, respectively. (c) Next, add points with smallest residual (blue points) into the subset and recompute PCA to the updated subset, where the result after five iterations is shown. (d) The final principal axes of the forward search is shown. In (d), the remaining points (black points) are regarded as outliers or minor regions, and the final principal axes are defined using the major regions (blue and red points).

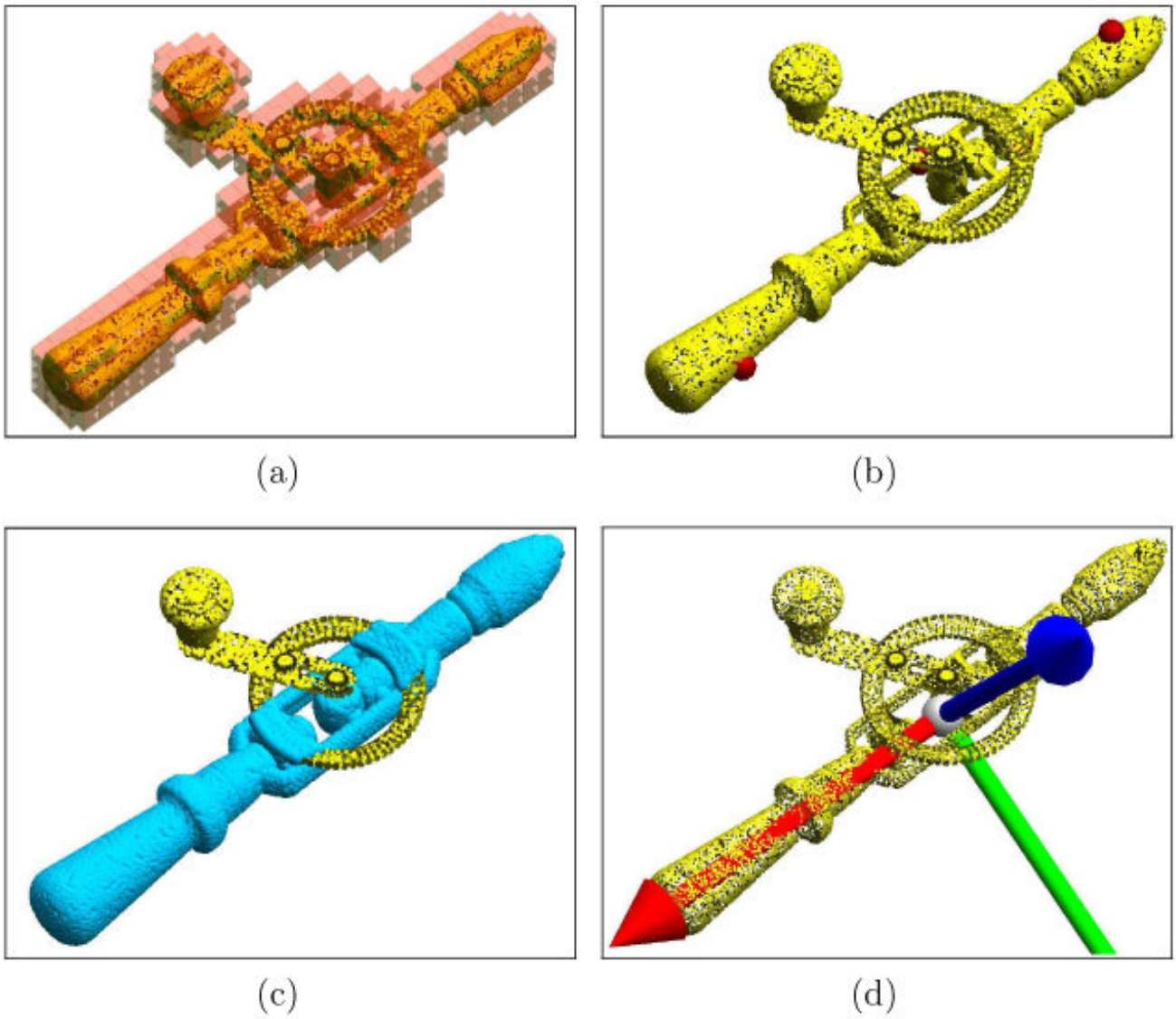

Fig. 2.
The illustration of robust principal axes determination for a point set representing the drill hand model with 23,400 points. (a) First, an octree of depth $d = 5$ is constructed with. (b) Next, the initial subset with 4 points (red) is chosen after 5000 iterations. (c) The final major region (blue points) is shown using the forward search technique. (d) The final principal axes are determined using the points on the major region, In this paper, the red, green, and blue axes correspond to the first, second, and third principal axes, respectively.

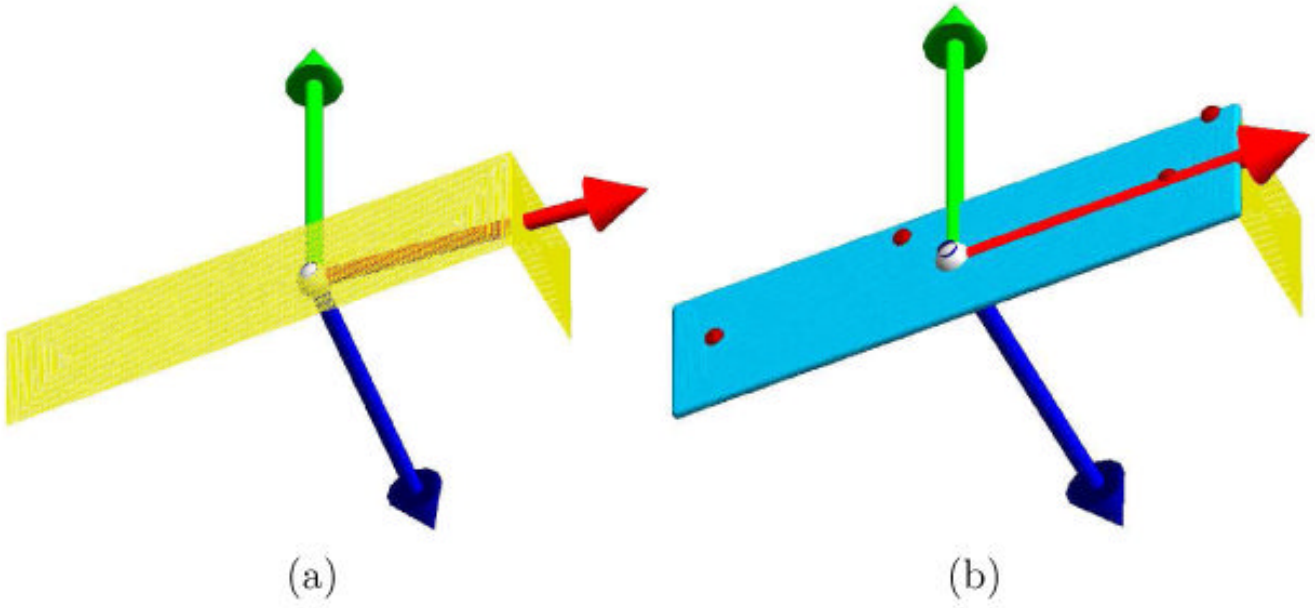

Fig. 3.

Determining principal axes for point sets sampled from surface patches. The input is a wedge data with 14,687 points sampled from two planes. (a) PCA. (b) Our method. Here four red points are the initial subset and blue points are major region. Note that the points sampled from the small plane do not effect our method unlike PCA.

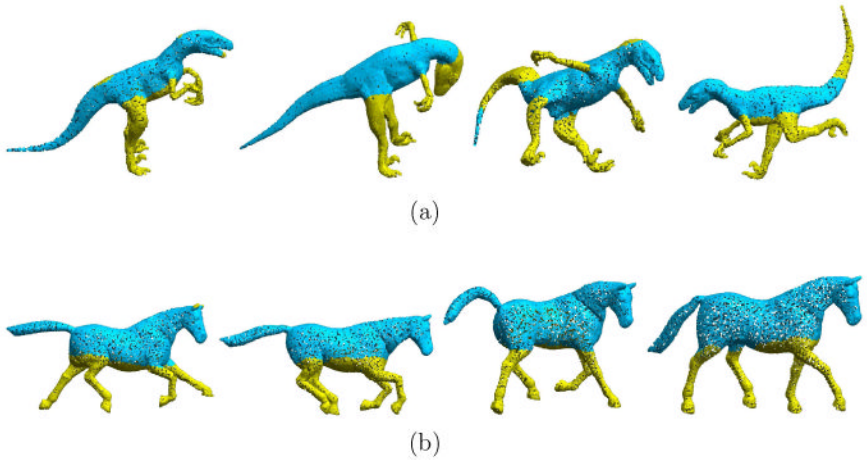

Fig. 4.
Testing the major regions (blue points) of similar models using our method. (a) One dinosaur with different poses. (b) One horse with different poses.

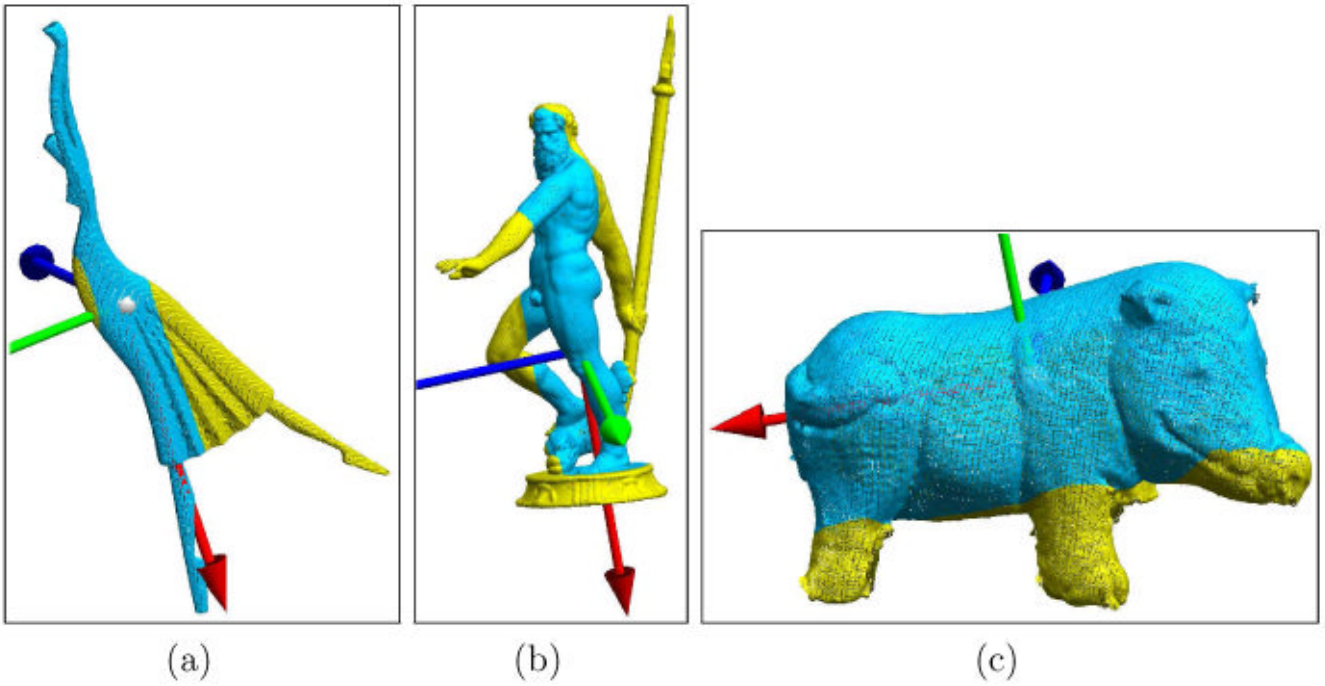

Fig. 5.
The major regions and principal axes of the scanned models using our method. (a) One dancer model. (b) Another dancer model with different pose. (c) The Neptune model. (d) The pig model.

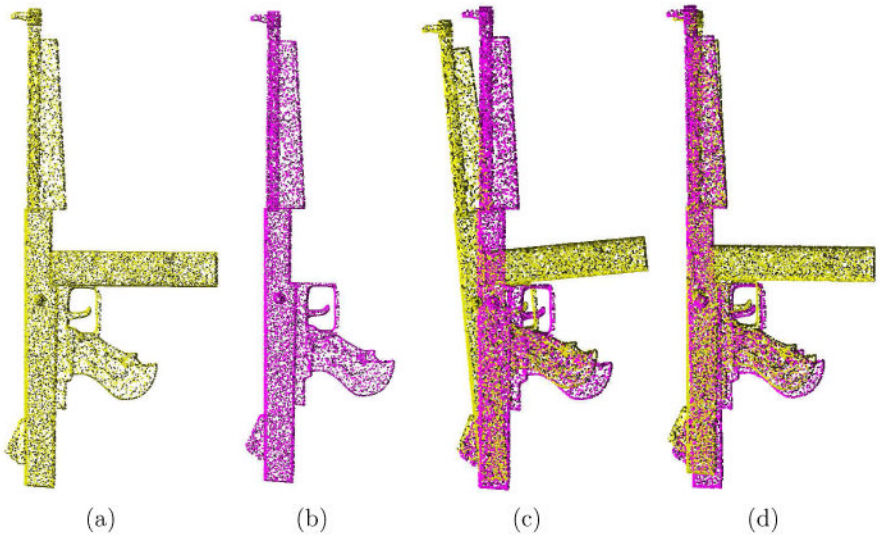

Fig. 6.

Shape alignment based on the global PCA and our method. (a) A point set sampled from a gun model. (b) Another point set sampled from the same gun, but without the cartridge clip. (c) The alignment result using PCA. (d) The alignment result using our method.

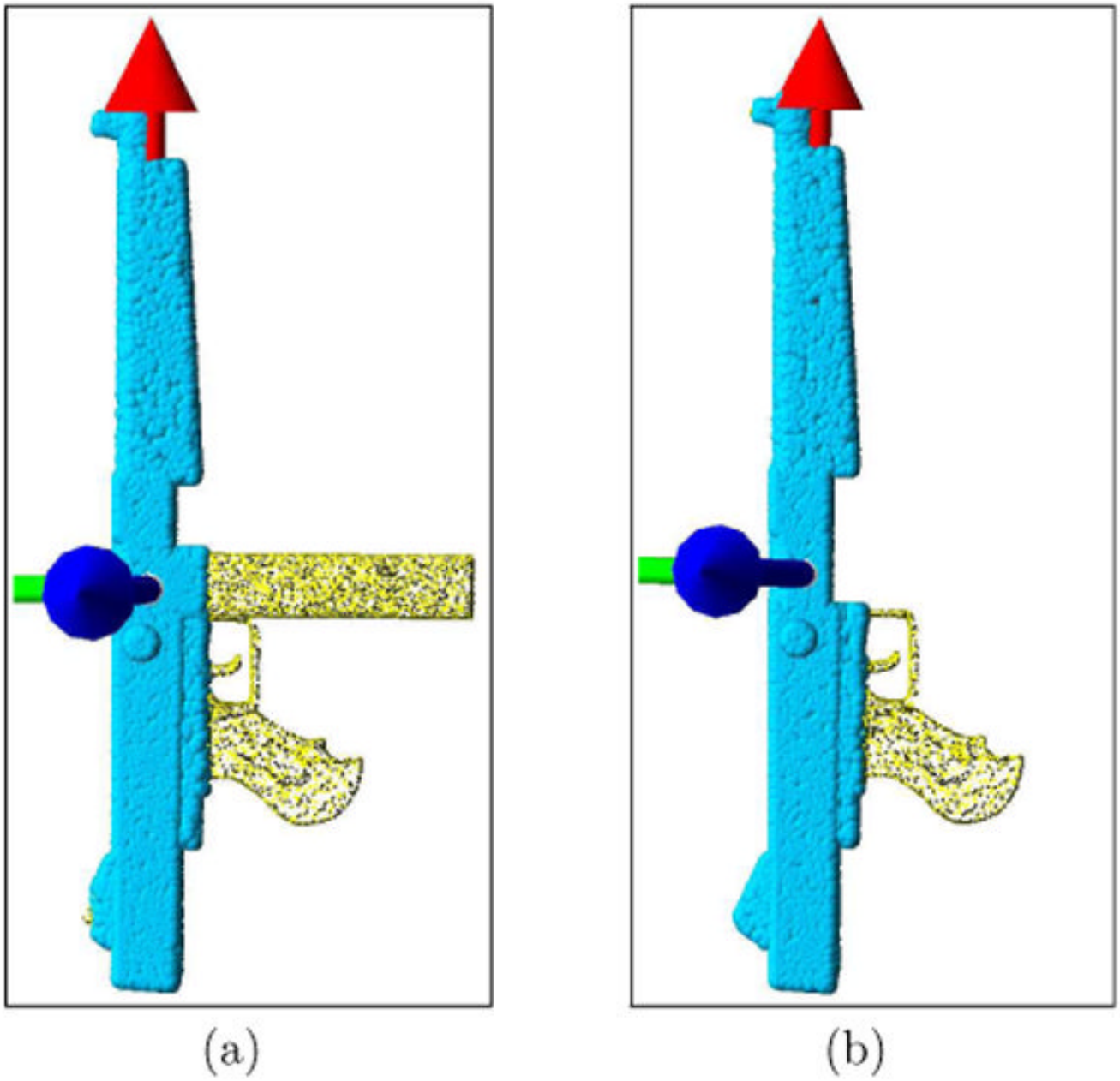

Fig. 7.
Comparison between two major regions (blue points) for Fig. 6(a) and Fig. 6(b) using our method. Here two major regions are almost consistent.

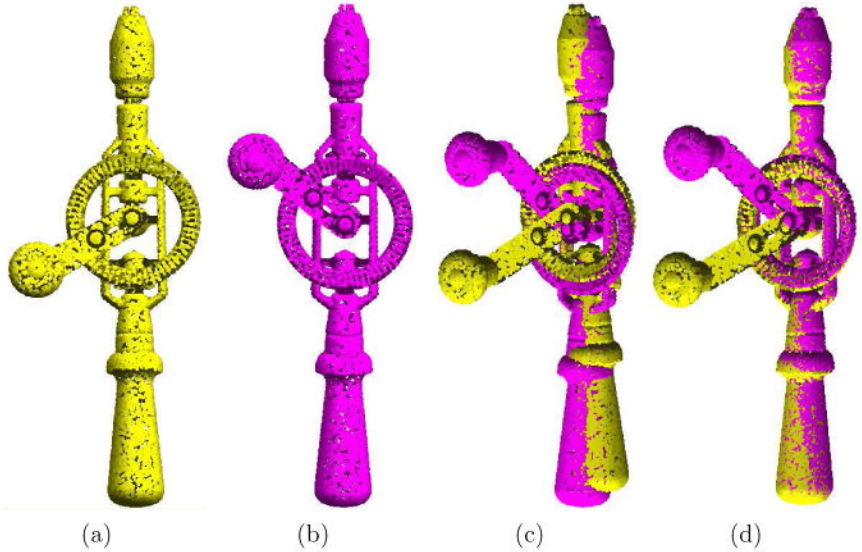

Fig. 8.
Aligning two models sampled from a drill with the different angle-rotation hand. (a) and (b) show the input data. (c) is the alignment result using PCA. (d) is the alignment result using our method.

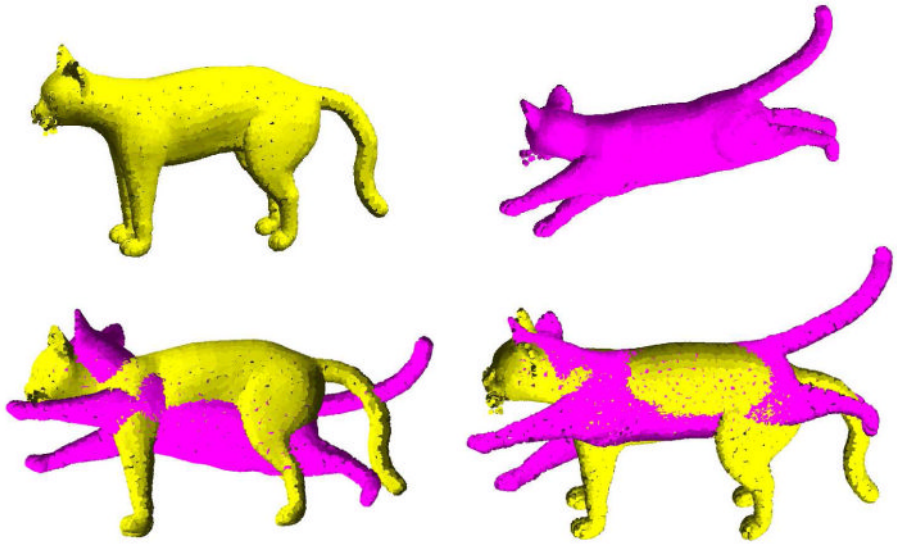

Fig. 9.
Aligning two models of a cat in different poses. In spite of the obvious difference between them, our algorithm aligns the two (bottom left) better than a global PCA alignment (bottom right).

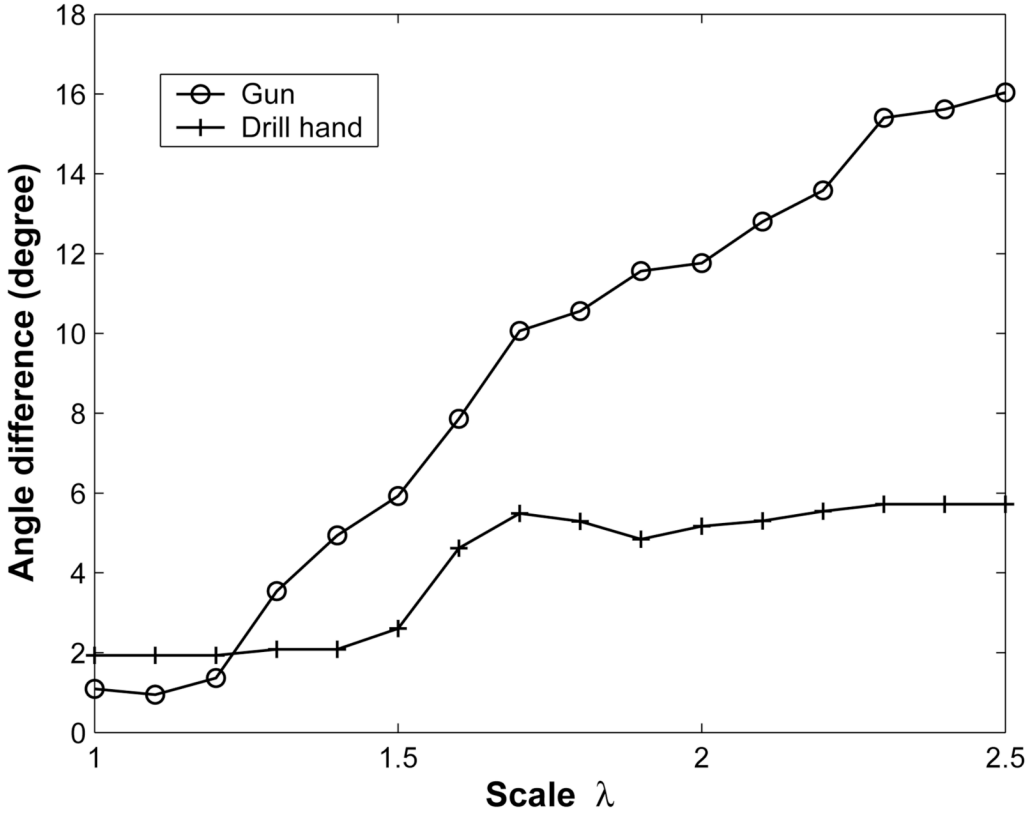

Fig. 10.

The angle difference between the initial and final first principal axes **e** with respect to the various scale λ of r_{max} in Eq. (6) for two models: gun and drill hand.

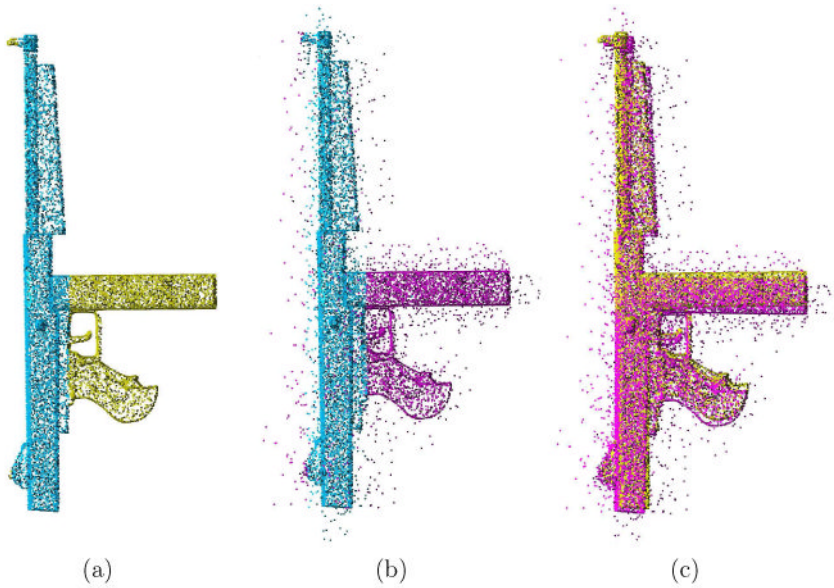

Fig. 11.

An alignment between the original model and the noisy. (a) The original point set and the computed major region (blue points). (a) The noisy point set and the computed major region (blue points). (c) The alignment result between the noisy model and the original one. Note that our method keeps the almost consistent major regions and principal axes between two models.

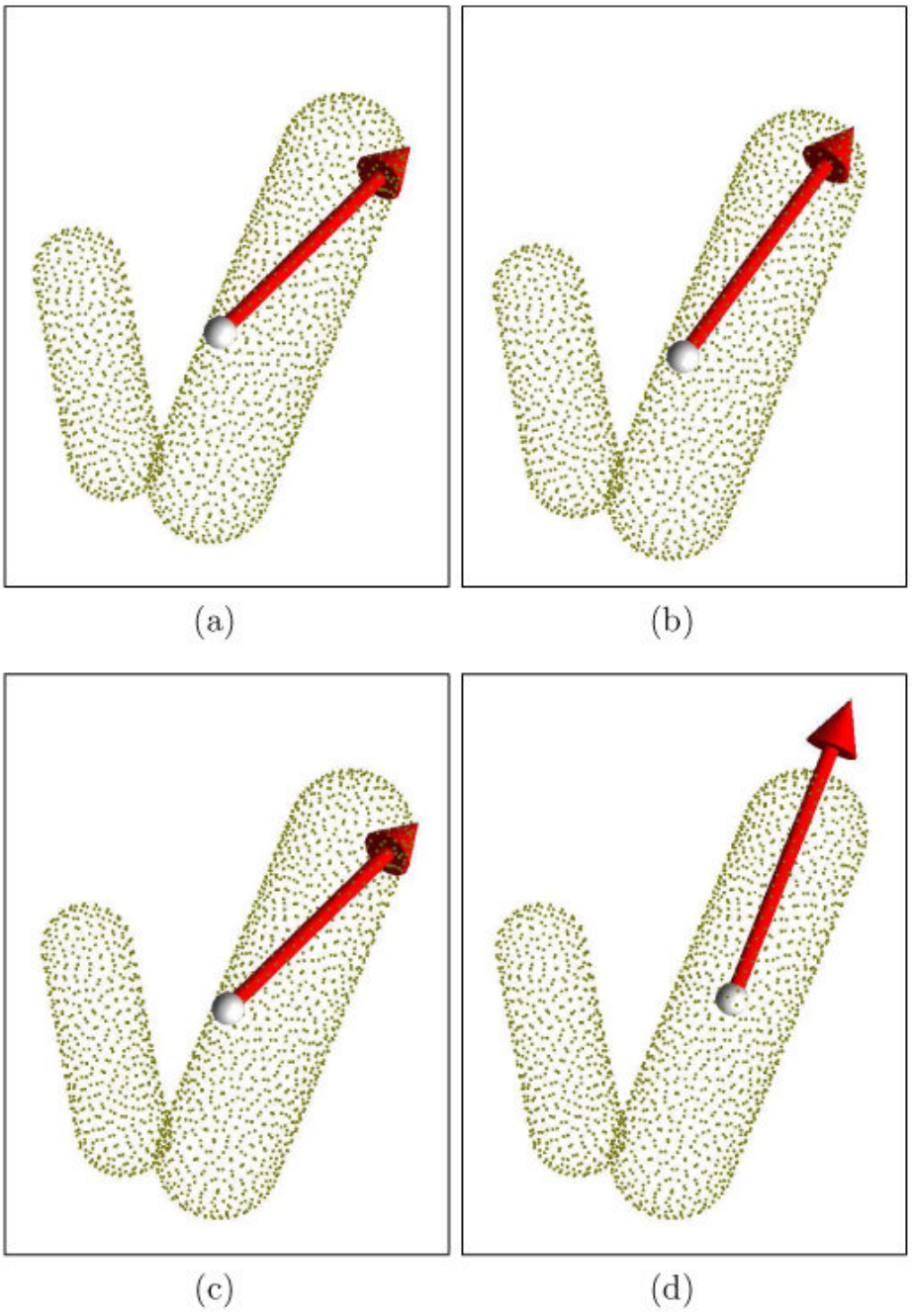

Fig. 12.

Comparison of four methods for determining the principal axis for a point set sampled from two objects. (a) PCA. (b) IRLS [39]. (c) LTS. (d) Our method. Note that the large object is identified as the major region for the final principal axis computation using our method. In contrast, PCA, IRLS and LTS are affected by the small object.

Table 1

Results of the **RobustPrincipalAxes** algorithm for some point-based shapes.

Model	**Fig.**	***N***	**Major%**	***m***	**T1** [a] **(s)**	**T2** [b] **(s)**
Drill	2	23,400	74.4%	60	0.07	9.32
Wedge	3(b)	14,687	80.3%	60	0.06	3.89
Gun	6(a)	15,412	66.5%	60	0.06	3.42
Cat	9	17,633	71.4%	60	0.07	7.91
Dancer	5(a)	75,206	63.8%	300	0.42	16.4
Neptune	5(b)	112,220	51.6%	300	0.28	21.3

[a] T1 is the time of constructing an octree at depth 5.

[b] T2 is the time of computing the principal axes.