J. Vis. Commun. Image R. 21 (2010) 442–452

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Scaling iterative closest point algorithm for registration of *m–D* point sets

Shaoyi Du [a,*], Nanning Zheng [a], Lei Xiong [b], Shihui Ying [c], Jianru Xue [a]

[a] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[b] School of Engineering, Air Force Engineering University, Xi'an 710049, China
[c] Department of Mathematics, School of Science, Shanghai University, Shanghai 200444, China

ARTICLE INFO

Article history:
Received 26 July 2009
Accepted 28 January 2010
Available online 23 February 2010

Keywords:
Iterative closest point (ICP)
Singular value decomposition (SVD)
Properties of parabola
Scaling registration
Point set registration
Iterative algorithm
Local convergence
Covariance matrix

ABSTRACT

Point set registration is important for calibration of multiple cameras, 3D reconstruction and recognition, etc. The iterative closest point (ICP) algorithm is accurate and fast for point set registration in a same scale, but it does not handle the case with different scales. This paper instead introduces a novel approach named the scaling iterative closest point (SICP) algorithm which integrates a scale matrix with boundaries into the original ICP algorithm for scaling registration. At each iterative step of this algorithm, we set up correspondence between two *m–D* point sets, and then use a simple and fast iterative algorithm with the singular value decomposition (SVD) method and the properties of parabola incorporated to compute scale, rotation and translation transformations. The SICP algorithm has been proved to converge monotonically to a local minimum from any given parameters. Hence, to reach desired global minimum, good initial parameters are required which are successfully estimated in this paper by analyzing covariance matrices of point sets. The SICP algorithm is independent of shape representation and feature extraction, and thereby it is general for scaling registration of *m–D* point sets. Experimental results demonstrate its efficiency and accuracy compared with the standard ICP algorithm.

1. Introduction

With the development of digital image, image registration especially feature-based image registration has become a key topic. When salient features of an image are represented as geometric entities, such as points, lines, and surfaces, it is of great importance to register image point sets with the optimal or suboptimal geometric transformation due to its wide applications such as calibration and synchronization of multiple cameras, 3D reconstruction using cameras. Moreover, as 3D measuring devices have been applied widely in recent years such as range scanner, stereoscopic plotter and medical equipment, it is crucial to register multi-view range images and build 3D geometric model for recognition.

In a word, point set registration has been a hot topic in computer vision, pattern recognition and image analysis. The iterative closest point (ICP) algorithm [1–3] is known for solving this problem for its good accuracy and fast speed. It has been widely used in a variety of fields such as 3D reconstruction [4], simultaneous localization and mapping [5], biometrics recognition [6], medical image analysis [7]. To speed up the traditional ICP algorithm, an increasing group of scholars have studied it. Fitzgibbon [8] employed the Levenberg–Marquardt algorithm to accelerate ICP, and Jost et al. [9] combined a coarse to fine multi-resolution technique with the neighbor search algorithm into ICP to improve the registration. Moreover, many scholars have introduced other methods into ICP for its more robustness. Lee et al. [10] proposed a measure for estimating the reliability of ICP. Invariant features were described by Sharp et al. [11] to decrease the probability of being trapped in a local minimum. Granger and Pennec [12] added Expectation-Maximization principles to ICP and used a coarse-to-fine approach based on an annealing scheme to improve the robustness. Silva et al. [13] introduced genetic algorithms and evaluation metric into ICP for more precise registration.

The original ICP algorithm does not take scale factor into account in the least squares (LS) problem, while in practice, the scale factor exists universally in registration since images acquired by real digitizers differ greatly in viewpoints and scanning resolutions, etc. Some scholars ever improved ICP for scaling registration. Zha et al. [14] used extended signature images to estimate the scale and applied it to traditional ICP for registration, while Zinßer et al. [15] directly estimated the scale in the ICP algorithm. Obviously the scale is a scalar that can only register two isotropic point sets, but not anisotropic ones which exists widely in scaling registration. To solve it, we have ever proposed to introduce a scale matrix directly into the LS problem with a constraint condition that the scale matrix is bounded [16]. The reason for adding this constraint condition is to avoid the phenomenon happening that

* Corresponding author. Fax: +86 29 83237910.
E-mail addresses: dushaoyi@gmail.com (S. Du), nnzheng@aiar.xjtu.edu.cn (N. Zheng), lei.xiong12@gmail.com (L. Xiong), yingshihui@gmail.com (S. Ying), jrxuester@gmail.com (J. Xue).

1047-3203/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2010.02.005

points of a set converge to a small subset of the other set. This paper details the scaling registration problem and the scaling iterative closest point (SICP) algorithm, an extension of the ICP algorithm to solve this constraint optimization problem. At each iterative step of the SICP algorithm, scale, rotation and translation transformations are computed through a new and fast iterative algorithm. Accordingly, the SICP algorithm has the similarly fast speed to the ICP algorithm and it also converges monotonically to a local minimum from any given parameters. To obtain the global minimum, we propose a simple method to estimate the initial parameters by analyzing eigenvalues and eigenvectors of covariance matrices of point sets. This new proposed algorithm has been tested in experiments and the experimental results demonstrate that our presented algorithm is a fast and robust technique to solve scaling registration problems caused by the scale factor, and it can be used widely in practice.

This paper is organized as follows. In Section 2, a general LS problem has been stated and the ICP algorithm has been reviewed briefly. In Section 3, an optimization problem is described with a constraint condition that the scale matrix is bounded, and a proposed method – the SICP algorithm is given. In Section 4, convergence theorems are given and initial parameters are discussed. Following that is Section 5 in which the proposed technique is evaluated on the experiments and a conclusion is finally drawn in the last section.

2. Problem statement and the ICP algorithm

2.1. Problem formulation

The registration of m–D point sets is a difficult problem. In this part, we formulate the point set registration as a general optimization problem. Given two point sets in $\mathbb{R}^m$, a model shape $M \triangleq \{\vec{m}_i\}_{i=1}^{N_m}$ and a data shape $P \triangleq \{\vec{p}_i\}_{i=1}^{N_p}$, $(N_m, N_p \in \mathbb{N})$, and assume that they have large overlap. A match between two m–D point sets is to find a transformation T, with which P is registered to be in best alignment with M, so the formulation is based on the following LS problem:

$$\min_{T, j\in\{1,2,\ldots,N_m\}} \left(\sum_{i=1}^{N_p} \left\| T(\vec{p}_i) - \vec{m}_j \right\|_2^2 \right) \tag{1}$$

This objective function represents a more general registration between two m–D point sets. In this function, T can be various transformations.

2.2. The ICP algorithm

The ICP algorithm proposed by Besl and McKay [1] is an efficient method to tackle rigid registration between two point sets. Its goal is to find a rigid transformation, with which P is registered to be in the best alignment with M, that is, let T of Eq. (1) be rotation and translation transformations, hence the rigid registration between two point sets is

$$\begin{aligned} \min_{\mathbf{R},\vec{t},j\in\{1,2,\ldots,N_m\}} \quad & \left(\sum_{i=1}^{N_p} \left\| (\mathbf{R}\vec{p}_i + \vec{t}) - \vec{m}_j \right\|_2^2 \right) \\ \text{s.t.} \quad & \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}_m, \ \det(\mathbf{R}) = 1 \end{aligned} \tag{2}$$

where $\mathbf{R} \in \mathbb{R}^{m\times m}$ is a rotation matrix, $\vec{t} \in \mathbb{R}^m$ is a translation vector.

The ICP algorithm achieves registration with good accuracy and fast speed, and it has two steps.

Firstly, set up correspondence between two point sets:

$$c_k(i) = \underset{j\in\{1,2,\ldots,N_m\}}{\arg\min} \left(\left\| (\mathbf{R}_{k-1}\vec{p}_i + \vec{t}_{k-1}) - \vec{m}_j \right\|_2^2 \right) \quad \text{for } i = 1,\ldots,N_p \tag{3}$$

Secondly, compute a new transformation between two point sets $\{\mathbf{R}_{k-1}\vec{p}_i + \vec{t}_{k-1}\}_{i=1}^{N_p}$ and $\{\vec{m}_{c_k(i)}\}_{i=1}^{N_p}$ by minimizing squared distance:

$$(\mathbf{R}^*, \vec{t}^*) = \underset{\mathbf{R}^{\mathrm{T}}\mathbf{R}=\mathbf{I}_m, \det(\mathbf{R})=1, \vec{t}}{\arg\min} \left(\sum_{i=1}^{N_p} \left\| \mathbf{R}(\mathbf{R}_{k-1}\vec{p}_i + \vec{t}_{k-1}) + \vec{t} - \vec{m}_{c_k(i)} \right\|_2^2 \right) \tag{4}$$

Update $\mathbf{R}_k$ and $\vec{t}_k$:

$$\mathbf{R}_k = \mathbf{R}^*\mathbf{R}_{k-1}, \quad \vec{t}_k = \mathbf{R}^*\vec{t}_{k-1} + \vec{t}^* \tag{5}$$

3. The SICP algorithm

3.1. The SICP algorithm

It is known that the ICP algorithm is a fast and accurate approach for rigid registration between two point sets. However, the scale factor may exist in point sets. In practice, we always need to consider the scaling registration, which includes scale transformation and rigid transformation:

$$\vec{x}' = T(\vec{x}) = \mathbf{RS}\vec{x} + \vec{t} \tag{6}$$

where $\mathbf{S} = diag(s_1, s_2, \ldots, s_m)$ is a scale matrix, $\mathbf{R}$ is an orthogonal matrix, and $\vec{t}$ is a translation vector.

If we now substitute Eq. (6) in Eq. (1), the function can be formulated as

$$\min_{\mathbf{S},\mathbf{R},\vec{t},j\in\{1,2,\ldots,N_m\}} \left(\sum_{i=1}^{N_p} \left\| (\mathbf{RS}\vec{p}_i + \vec{t}) - \vec{m}_j \right\|_2^2 \right) \tag{7}$$

The LS problem which is presented in Eq. (7) is to deal with the scaling registration. In Eq. (7), the orthogonal matrix $\mathbf{R}$ can denote reflection and rotation transformations. As reflection and rotation transformations are two independent transformations; meanwhile, reflection transformation is already given at the initial stage of the registration, what needs to be considered in this paper is rotation transformation only. Moreover, the scale matrix $\mathbf{S}$ must be bounded in practice. The reason for this is to avoid the phenomenon happening that points of one set converge to a small subset of the other set, namely, the scale matrix is close to $\mathbf{0}$. With the exposition above, the registration problem now becomes the following LS problem:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{R},\vec{t},j\in\{1,2,\ldots,N_m\}} \quad & \left(\sum_{i=1}^{N_p} \left\| (\mathbf{RS}\vec{p}_i + \vec{t}) - \vec{m}_j \right\|_2^2 \right) \\ \text{s.t.} \quad & \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}_m, \ \det(\mathbf{R}) = 1 \\ & \mathbf{S} = diag(s_1, s_2, \ldots, s_m), \ s_j \in [a_j, b_j] \end{aligned} \tag{8}$$

where $\mathbf{S}$ is a scale matrix with boundary and $\mathbf{R}$ is a rotation matrix.

How to solve Eq. (8) is the focus of this paper. Actually, we can solve this problem in the way the ICP algorithm does by iteration. At each iteration, two steps are included:

Step 1: Set up correspondence by the current transformation $(\mathbf{S}_{k-1}, \mathbf{R}_{k-1}, \vec{t}_{k-1})$:

$$c_k(i) = \underset{j\in\{1,2,\ldots,N_m\}}{\arg\min} \left(\left\| (\mathbf{R}_{k-1}\mathbf{S}_{k-1}\vec{p}_i + \vec{t}_{k-1}) - \vec{m}_j \right\|_2^2 \right) \tag{9}$$

Step 2: Assume $\mathbf{S} = diag(s_1, s_2, \ldots, s_m)$, compute a new transformation $(\mathbf{S}_k, \mathbf{R}_k, \vec{t}_k)$:

$$(\mathbf{S}_k, \mathbf{R}_k, \vec{t}_k) = \underset{s_j\in[a_j,b_j],\mathbf{R},\vec{t}}{\arg\min} \left(\sum_{i=1}^{N_p} \left\| \mathbf{RS}\vec{p}_i + \vec{t} - \vec{m}_{c_k(i)} \right\|_2^2 \right) \tag{10}$$

3.2. Computation of scale, rotation and translation transformations

Step 1 of the SICP algorithm can be solved by many methods such as the nearest point search algorithm based on Delaunay triangulations [17]; hence step 2 is the key step. To compute the new scale, rotation and translation transformations, the following lemma is given first to eliminate the translation transformation:

Lemma 1. *Given two m–D point sets* $\{\vec{q}_i\}_{i=1}^N$ *and* $\{\vec{n}_i\}_{i=1}^N$, *the function* $F(\vec{t}) = \sum_{i=1}^N \|\vec{q}_i + \vec{t} - \vec{n}_i\|_2^2$ *has the minimum when* $\vec{t} = \frac{1}{N}\sum_{i=1}^N \vec{n}_i - \frac{1}{N}\sum_{i=1}^N \vec{q}_i$.

Proof. If $F(\vec{t})$ reaches the minimum, it must satisfy the following requirement:

$$\frac{dF(\vec{t})}{d\vec{t}} = 0$$

As $\frac{dF(\vec{t})}{d\vec{t}} = 2\sum_{i=1}^N (\vec{q}_i + \vec{t} - \vec{n}_i) = 0$, we can obtain $\vec{t} = \frac{1}{N}\sum_{i=1}^N \vec{n}_i - \frac{1}{N}\sum_{i=1}^N \vec{q}_i$. □

According to Lemma 1, if minimizing $F(\mathbf{S},\mathbf{R}) = \sum_{i=1}^{N_p} \|(\mathbf{RS}\vec{p}_i + \vec{t}) - \vec{m}_{c_k(i)}\|_2^2$, we get $\vec{t} = \frac{1}{N_p}\sum_{i=1}^{N_p} \vec{m}_{c_k(i)} - \frac{1}{N_p}\sum_{i=1}^{N_p} \mathbf{RS}\vec{p}_i$. Hence,

$$F(\mathbf{S},\mathbf{R}) = \sum_{i=1}^{N_p} \left\| \mathbf{RS}\left(\vec{p}_i - \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{p}_i\right) - \left(\vec{m}_{c_k(i)} - \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{m}_{c_k(i)}\right)\right\|_2^2$$

Let $\vec{q}_i \triangleq \vec{p}_i - \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{p}_i$ and $\vec{n}_i \triangleq \vec{m}_{c_k(i)} - \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{m}_{c_k(i)}$, therefore,

$$\begin{aligned} F(\mathbf{S},\mathbf{R}) &= \sum_{i=1}^{N_p} \|\mathbf{RS}\vec{q}_i - \vec{n}_i\|_2^2 \\ &= \sum_{i=1}^{N_p} \vec{q}_i^T\mathbf{S}^2\vec{q}_i - 2\sum_{i=1}^{N_p}\vec{n}_i^T\mathbf{RS}\vec{q}_i + \sum_{i=1}^{N_p}\vec{n}_i^T\vec{n}_i \end{aligned} \tag{11}$$

To minimize Eq. (11), we can recover the following partial differential equations to compute scale and rotation transformations:

$$\frac{\partial F(\mathbf{S},\mathbf{R})}{\partial \mathbf{R}} = 0 \tag{12}$$

$$\frac{\partial F(\mathbf{S},\mathbf{R})}{\partial \mathbf{S}} = 0 \tag{13}$$

3.2.1. Rotation computation

The rigid registration between two paired point sets is a well-studied problem in the literature. Many closed-form methods are known to be used to compute the rigid transformation: singular value decomposition (SVD) [18], unit quaternions [19], orthonormal matrices [20] and dual quaternions [21]. An overview of these methods and a comparative analysis can be found in [22]. As the SVD algorithm still works even when **S** is considered, we will detail the method to compute rotation transformation of Eq. (12) as follows.

For any given **S**, the necessary condition of minimizing $F(\mathbf{S},\mathbf{R})$ is Eq. (12) which cannot be computed easily, but according to Eq. (11), minimizing $F(\mathbf{S},\mathbf{R})$ is equivalent to maximizing $\sum_{i=1}^{N_p}\vec{n}_i^T\mathbf{RS}\vec{q}_i$, which can be solved similar to that Arun [18] had proposed, thus we only give the conclusion here.

1. Calculate $m \times m$ matrix **H** and its SVD.

$$\mathbf{H} = \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{S}\vec{q}_i\vec{n}_i^T \tag{14}$$

$$\mathbf{H} = \mathbf{U}\Lambda\mathbf{V} \tag{15}$$

2. Calculate the rotation matrix **R**.
   (a) If $\det(\mathbf{VU}^T) = +1$, $\mathbf{VU}^T$ is a rotation:

$$\mathbf{R} = \mathbf{VU}^T \tag{16}$$

   (b) If $\det(\mathbf{VU}^T) = -1$, $\mathbf{VU}^T$ is a reflection:

   (i) If one of the singular values of **H** is zero, the desired rotation can be calculated as follows:

$$\mathbf{R} = \mathbf{V}\begin{pmatrix} \mathbf{I}_{m-1} & 0 \\ 0 & -1 \end{pmatrix}\mathbf{U}^T \tag{17}$$

   (ii) If none of the singular values of **H** is zero, we go to a RANSAC-like technique.

3.2.2. Scale computation

Suppose that $\mathbf{E}_j = diag(0,\ldots,0,1,0,\ldots,0)$, $(j = 1,2,\ldots,m)$ is a diagonal matrix where the j^{th} element is one, but others are zero. They are the basis of the matrix **S**, and then Eq. (13) can be expressed as follows:

$$\begin{aligned} \frac{\partial F(\mathbf{S},\mathbf{R})}{\partial \mathbf{S}} &= \lim_{t\to 0}\frac{F(\mathbf{S}+t\mathbf{E}_j,\mathbf{R}) - F(\mathbf{S},\mathbf{R})}{t} \\ &= 2\sum_{i=1}^{N_p}\vec{q}_i^T\mathbf{S}\mathbf{E}_j\vec{q}_i - 2\sum_{i=1}^{N_p}\vec{n}_i^T\mathbf{R}\mathbf{E}_j\vec{q}_i = 0 \end{aligned} \tag{18}$$

From Eq. (18), we get

$$s_j = \frac{\sum_{i=1}^{N_p}\vec{n}_i^T\mathbf{R}\mathbf{E}_j\vec{q}_i}{\sum_{i=1}^{N_p}\vec{q}_i^T\mathbf{E}_j\vec{q}_i} \tag{19}$$

1. If s_j is any arbitrary number, we obtain the scale of SICP with unbounded scale:

$$s_j = \sum_{i=1}^{N_p}\vec{n}_i^T\mathbf{R}\mathbf{E}_j\vec{q}_i \Big/ \sum_{i=1}^{N_p}\vec{q}_i^T\mathbf{E}_j\vec{q}_i \tag{20}$$

2. If $s_j \in [a_j, b_j]$, according to Eq. (11), the function is known to be a parabola with respect to s_j and its symmetry axis parallels to vertical axis, so the minimum can be achieved at the point which is nearest to the vertex of this parabola, hence we get the scale s_j of the SICP algorithm:

$$s_j = \arg\min_{s\in[a_j,b_j]}\left| s - \sum_{i=1}^{N_p}\vec{n}_i^T\mathbf{R}\mathbf{E}_j\vec{q}_i \Big/ \sum_{i=1}^{N_p}\vec{q}_i^T\mathbf{E}_j\vec{q}_i \right| \tag{21}$$

3.2.3. Termination

After the computation of rotation and scale transformations above, we check whether the change of **S** is less than ε or a maximum number of iterations is reached. If it does, we switch outside the loop, otherwise we continue the iteration.

3.2.4. Translation computation

If rotation and scale transformations are computed, according to Lemma 1, we calculate $\vec{t}_k$:

$$\vec{t}_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{m}_{c_k(i)} - \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{R}_k\mathbf{S}_k\vec{p}_i \tag{22}$$

From what is discussed above, the scale, rotation and translation transformations of Eq. (10) are computed. Therefore, the whole SICP algorithm is reasonably drawn out as follows.

Input: Two point sets $P \triangleq \{\vec{p}_i\}_{i=1}^{N_p}$ and $M \triangleq \{\vec{m}_i\}_{i=1}^{N_m}$.
Initialize: Scale matrix $\mathbf{S}_0$, rotation matrix $\mathbf{R}_0$, translation vector $\vec{t}_0$ and the scale boundary are initialized by using covariance matrices of point sets, which is stated in Section 4.

Repeat

Step 1: Compute Eq. (9) to set up correspondence $\{(\vec{p}_i, \vec{m}_{c_k(i)})\}_{i=1}^{N_p}$ by the $(k-1)^{th}$ transformation $(\mathbf{S}_{k-1}, \mathbf{R}_{k-1}, \vec{t}_{k-1})$.

Step 2: Let $\mathbf{S}_{k,0} = \mathbf{S}_{k-1}$, $\mathbf{R}_{k,0} = \mathbf{R}_{k-1}$, $\vec{q}_i \triangleq \vec{p}_i - \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{p}_i$ and $\vec{n}_i \triangleq \vec{m}_{c_k(i)} - \frac{1}{N_p}\sum_{i=1}^{N_p}\vec{m}_{c_k(i)}$.

Repeat

1: For any given $\mathbf{S}_{k,n-1}$, $\mathbf{R}_{k,n}$ can be calculated according to Eqs. (14)–(17) which minimizes $F(\mathbf{S}_{k,n-1}, \mathbf{R})$ and satisfies the condition $\mathbf{R}_{k,n}^T\mathbf{R}_{k,n} = \mathbf{I}_m$, $\det(\mathbf{R}_{k,n}) = 1$.

2: For any given $\mathbf{R}_{k,n}$, $\mathbf{S}_{k,n}$ can be calculated according to Eqs. (20) or (21) which minimizes $F(\mathbf{S}, \mathbf{R}_{k,n})$ and satisfies the condition $\mathbf{S}_{k,n} \in diag([a_1, b_1], \ldots, [a_m, b_m])$.

until $\|\mathbf{S}_{k,n} - \mathbf{S}_{k,n-1}\| \leqslant \varepsilon$ or n reaches a maximum number of iterations.

Let $\mathbf{S}_k = \mathbf{S}_{k,n}$ and $\mathbf{R}_k = \mathbf{R}_{k,n}$, and calculate $\vec{t}_k$ by Eq. (22).

until $|\varepsilon(\mathbf{S}_k, \mathbf{R}_k, \vec{t}_k) - \varepsilon(\mathbf{S}_{k-1}, \mathbf{R}_{k-1}, \vec{t}_{k-1})| \leqslant \varepsilon$ or k reaches a maximum number of iterations.

Output: Scale matrix $\mathbf{S}_k$, rotation matrix $\mathbf{R}_k$ and translation vector $\vec{t}_k$.

From the whole SICP algorithm, we know that the SICP algorithm is similar to the ICP algorithm, which is composed of two basic steps: to establish the correspondence and to compute the transformation. Although the SICP algorithm can not give an analytical solution for transformation parameters at each iteration, it gets similar computation time as the ICP algorithm for two reasons. One is that the computation time of the transformation is far less than that of the correspondence, so the computation time of both algorithms focuses on the computation time of the correspondence, both of which are same. The other reason is that as the data shape moves closer to the model shape, our algorithm needs rather fewer iterative steps for computation of the scale, rotation and translation transformations. Hence, the SICP algorithm is fast.

4. Convergence theorem and initial parameters

In this section, the convergence of the SICP algorithm will be stated and proved. SICP and ICP share the same procedure that the data shape searches the closest points in the model shape by iteration. Although the data shape cannot find the correct corresponding points in the model shape in one step, it moves to the correct point set much nearer by iteration. The following theorems will elaborately prove SICP converges in theory.

Theorem 1. *The algorithm for computing the scale and rotation matrices in* Section 3.2 *converges monotonically to a minimum.*

Proof. Given $\mathbf{S}_{k,n-1}$, from the algorithm we know $\mathbf{R}_{k,n}$ calculated in the step of rotation computation can minimize $F(\mathbf{S}_{k,n-1}, \mathbf{R})$ and satisfy the condition $\mathbf{R}_{k,n}^T\mathbf{R}_{k,n} = \mathbf{I}_m$, $\det(\mathbf{R}_{k,n}) = 1$. It is clear that $F(\mathbf{S}_{k,n-1}, \mathbf{R}_{k,n}) \leqslant F(\mathbf{S}_{k,n-1}, \mathbf{R}_{k,n-1})$.

Next, in the step of scale computation, $F(\mathbf{S}_{k,n}, \mathbf{R}_{k,n})$ is the minimum of $F(\mathbf{S}, \mathbf{R}_{k,n})$ and $\mathbf{S}_{k,n}$ satisfies the condition $\mathbf{S}_{k,n} \in diag([a_1, b_1], \ldots, [a_m, b_m])$, so it is easy to infer that $F(\mathbf{S}_{k,n}, \mathbf{R}_{k,n}) \leqslant F(\mathbf{S}_{k,n-1}, \mathbf{R}_{k,n})$.

Hence, repeat above procedures, we obtain

$$0 \leqslant \cdots \leqslant F(\mathbf{S}_{k,n}, \mathbf{R}_{k,n}) \leqslant F(\mathbf{S}_{k,n-1}, \mathbf{R}_{k,n}) \leqslant F(\mathbf{S}_{k,n-1}, \mathbf{R}_{k,n-1}) \leqslant \cdots \quad \text{for all } n$$

According to the Monotonic Sequence Theorem "Every bounded monotonic sequence of real numbers is convergent", the algorithm for computing the scale and rotation matrices converges monotonically to a minimum. □

Theorem 2. *The SICP algorithm converges monotonically to a local minimum with respect to square distance.*

Proof. Given two point sets $M \triangleq \{\vec{m}_i\}_{i=1}^{N_m}$ and $P \triangleq \{\vec{p}_i\}_{i=1}^{N_p}$. Denote $\mathbf{S}_k$, $\mathbf{R}_k$ and $\vec{t}_k$ as scale matrix, rotation matrix and translation vector, respectively. Let $\vec{q}_{i,k-1} \triangleq \mathbf{R}_{k-1}\mathbf{S}_{k-1}\vec{p}_i + \vec{t}_{k-1}$, then we can compute its corresponding closet point $\vec{m}_{c_k(i)}$ in the model shape. Therefore, the square distance is

$$e_k = \sum_{i=1}^{N_p} \left\|\vec{q}_{i,k-1} - \vec{m}_{c_k(i)}\right\|_2^2$$

Because $\{\vec{p}_i\}$ is to register $\{\vec{m}_{c_k(i)}\}$, we compute the best scale, rotation and translation transformations. Hence, the square distance is

$$\varepsilon_k = \sum_{i=1}^{N_p} \left\|\mathbf{S}_k\mathbf{R}_k\vec{p}_i + \vec{t}_k - \vec{m}_{c_k(i)}\right\|_2^2$$

It is obvious that $\varepsilon_k \leqslant e_k$. Otherwise, let $\mathbf{S}_k = \mathbf{S}_{k-1}, \mathbf{R}_k = \mathbf{R}_{k-1}, \vec{t}_k = \vec{t}_{k-1}$, we obtain $e_k = \varepsilon_k$. However, according to Theorem 1, ε_k is the minimum of Eq. (10), hence $\varepsilon_k \leqslant e_k$.

Next, suppose $\vec{q}_{i,k} \triangleq \mathbf{S}_k\mathbf{R}_k\vec{p}_i + \vec{t}_k$, then square distance is e_{k+1}, that is,

$$e_{k+1} = \sum_{i=1}^{N_p} \left\|\vec{q}_{i,k} - \vec{m}_{c_{k+1}(i)}\right\|_2^2$$

Because e_{k+1} is the minimum of Eq. (9), then

$$e_{k+1} = \sum_{i=1}^{N_p} \left\|\vec{q}_{i,k} - \vec{m}_{c_{k+1}(i)}\right\|_2^2 \leqslant \sum_{i=1}^{N_p} \left\|\vec{q}_{i,k} - \vec{m}_{c_k(i)}\right\|_2^2 = \varepsilon_k$$

Hence, repeat procedures above, we obtain

$$0 \leqslant \cdots \leqslant e_{k+1} \leqslant \varepsilon_k \leqslant e_k \leqslant \cdots \quad \text{for all } k$$

According to the Monotonic Sequence Theorem "Every bounded monotonic sequence of real numbers is convergent", the SICP algorithm converges monotonically to a local minimum with respect to square distance. □

Theorem 2 shows the SICP algorithm converges monotonically to a local minimum for any given initial parameters. To obtain the global minimum, the usual way is to find the minimum of all local minima. However, it is difficult to characterize precisely and generally that the registration state space is partitioned into local minima regions, for the encountered shapes are quite different. Therefore, how to estimate initial parameters well and give the constraints is difficult. This paper discusses some cases about the initial parameters of scale and rotation transformations with corresponding constraints. Since rotation parameter has been detailed in [1,10] by using covariance matrices, this paper will extend the works and discuss some cases about the initial rotation and scale parameters with corresponding constraints briefly.

Variability is able to be described by covariance matrix from which eigenvectors and eigenvalues are calculated. The eigenvector corresponding to the largest eigenvalue denotes the direction of largest variance of the data set and the eigenvectors are orthonormal. When shapes will not deform greatly, that is, the ratio of any two scales is not large, the eigenvalues and eigenvectors of covariance matrices are used to estimate the initial parameters and the boundary of the scale parameter.

Assume that points are sampled enough from two shapes and their covariance matrices are $\mathbf{C}_M$ and $\mathbf{C}_P$. The square roots of $\mathbf{C}_M$ and $\mathbf{C}_P$'s eigenvalues are $\lambda_1, \ldots, \lambda_m$ and $\mu_1, \ldots, \mu_m$, respectively, with corresponding eigenvectors $p_1, \ldots, p_m$ and $r_1, \ldots, r_m$. When one shape is deformed with a scale matrix, the eigenvectors may deviate the original axes and the eigenvalues may change large

Table 1
Compared results on two 2D shapes with the same scale.

Point sets	ICP		SICP		SICP with unbounded scale	
	Scale	RMS	Scale	RMS	Scale	RMS
Bat	1.0000	0.5608	*diag*(1.0005, 1.0009)	0.5518	*diag*(0.5505, 0.3570)	34.8171
Butterfly	1.0000	0.5663	*diag*(1.0006, 1.0012)	0.5640	*diag*(0.4490, 0.7549)	16.7619
Horse	1.0000	0.7783	*diag*(1.0003, 1.0003)	0.7779	*diag*(0.1064, 0.8396)	5.1919

or less. So is the case when some noise or missing data adds to one shape. Hence, the initial rotation matrix is the relation of any two eigenvectors between two shapes rather than being obtained by setting up the relation of eigenvectors with sorted corresponding eigenvalues. Moreover, because of the bidirection of eigenvectors, the total number of the initial rotation matrices is $2m \times 2(m-1) \times \cdots \times 2 = 2^m m!$, which is quite a lot. As the SICP algorithm has wide large convergence domain because of its bounded scale, it is not necessary at all most of time to try all rotation matrices and the rotation matrix is initialized to be the identity transformation or the 180° rotation of all eigenvector axes in this paper.

When the data shape P registers the model shape M, the eigenvalues of data shape will scale to the corresponding eigenvalues of model shape. Because of the noise, missing data and the anisotropic scale factor, the corresponding eigenvector axes are ambiguous while the average value of the ratios of eigenvalues is well for the scale initialization. Hence, the initial scale matix $\mathbf{S}_0 = diag(s_{01}, s_{02}, \ldots, s_{0m})$ and its constraint are estimated as follows:

$$s_{0i} = \frac{1}{m}\sum_{j=1}^{m}\frac{\mu_j}{\lambda_j} \quad (i = 1, 2, \ldots, m) \tag{23}$$

$$\frac{1}{m}\sum_{j=1}^{m}\frac{\lambda_j}{\mu_j} - \delta \leqslant s_{0i} \leqslant \frac{1}{m}\sum_{j=1}^{m}\frac{\lambda_j}{\mu_j} + \delta \tag{24}$$

where δ is a given value meaning the tolerance of the deformation, which can be given manually or by other methods. In this paper, assume η is $\frac{1}{m}\sum_{j=1}^{m}\frac{\mu_j}{\lambda_j}$. When the relative scaling is small, we set $\delta = 0.1\eta$.

5. Experimental results

To verify the robustness and convergence of our presented method, experiments are tested on the following data sets: (1) certain 2D shapes in part B of CE-Shape-1 [23], (2) the Stanford 3D Scanning Repository.[1] The results of the ICP and SICP algorithms are reported as follows in which errors are computed by root mean square (RMS). All programs are written in Matlab 7.0 and are run on PC with Pentium 4 3.6 G CPU and 1 G RAM.

5.1. 2D shapes matching

The following experiments are conducted to test our algorithm's robustness and efficiency for registration of 2D shapes. We use some shapes of Part B of CE-Shape-1, a large 2D shapes database, to compare ICP and SICP. In each experiment, one shape is selected as the data shape, and the other the model shape. In the experiments, we extract the edges of 2D shapes as point sets and use covariance matrices of these point sets to estimate the scale parameter and its constraint in the SICP algorithm. And to prove the scale of SICP needs to be bounded, we also give compared results of SICP with unbounded scale, the initial parameters of which are estimated by covariance matrices as well.

1. First, we prove SICP is similar to ICP which can complete registration of 2D shapes with the same scale. In SICP, we estimate the initial scale matrix $\mathbf{S}_0$ and its constraint from Eqs. (23) and (24), and the compared results are shown in Table 1.

Table 1 reveals SICP can have almost same accuracy as ICP because RMS and scale of ICP and SICP are similar. Furthermore, RMS of SICP with unbounded scale is much larger than that of the other two algorithms and its coordinate is much smaller, which means that SICP with unbounded scale produces unsatisfied

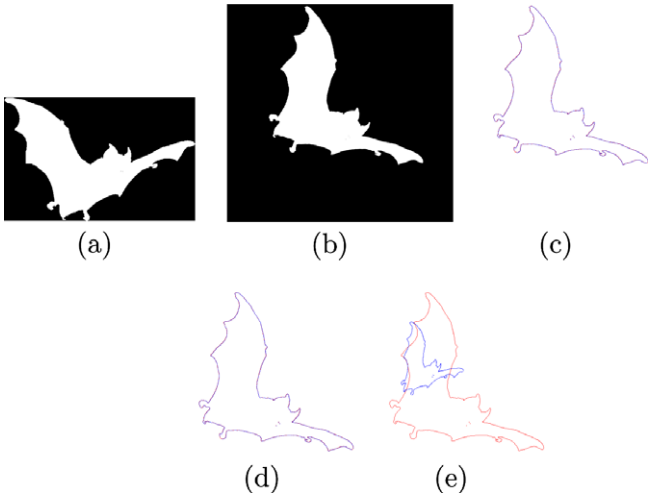

Fig. 1. Registration results on 2D bat shapes. (a) 2D data shape. (b) 2D model shape. (c) Registration result of ICP. (d) Registration result of SICP. (e) Registration result of SICP with unbounded scale.

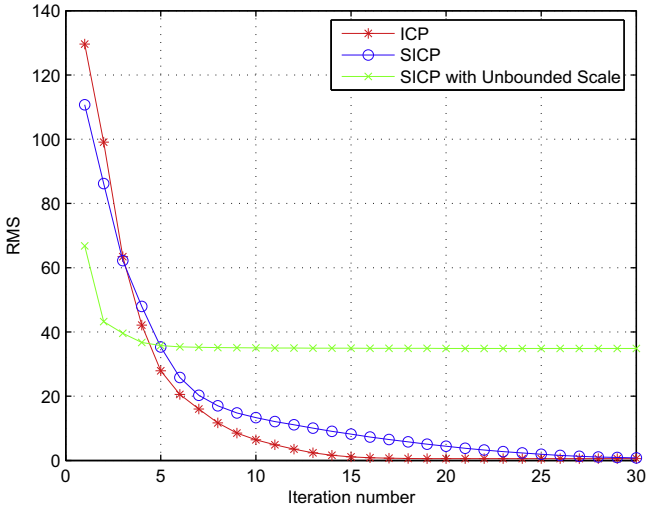

Fig. 2. The convergence of ICP, SICP and SICP with unbounded scale on 2D bat shapes.

[1] http://graphics.stanford.edu/data/3Dscanrep/.

Table 2
Compared results on two 2D shapes with different scales.

Point sets	ICP		SICP		SICP with unbounded scale	
	Scale	RMS	Scale	RMS	Scale	RMS
Beetle	1.0000	19.8742	*diag*(1.7529, 1.7523)	0.9628	*diag*(1.5598, 0.0022)	11.8789
Bell	1.0000	17.6304	*diag*(1.5889, 1.6922)	5.4280	*diag*(1.5889, 1.6922)	5.4280
Turtle	1.0000	11.3636	*diag*(1.0852, 1.1974)	7.7115	*diag*(0.0000, 0.9694)	5.4695

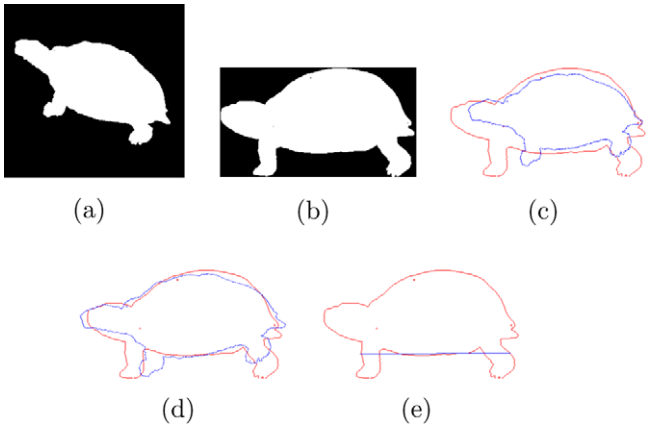

Fig. 3. Registration results on 2D turtle shapes. (a) 2D data shape. (b) 2D model shape. (c) Registration result of ICP. (d) Registration result of SICP. (e) Registration result of SICP with unbounded scale.

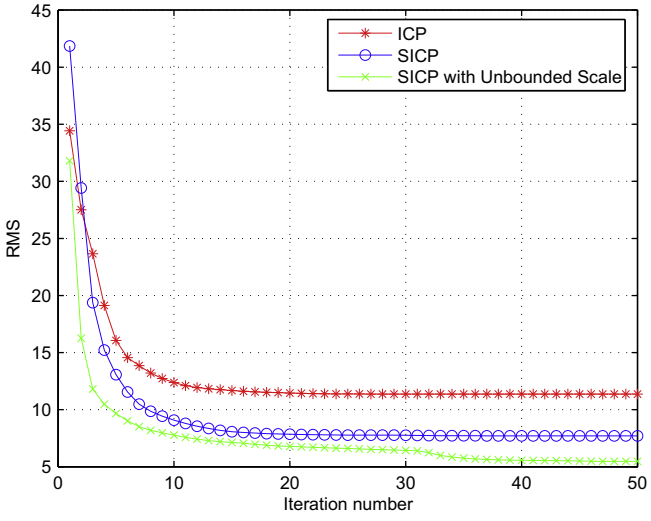

Fig. 4. The convergence of ICP, SICP and SICP with unbounded scale on 2D turtle shapes.

results. The typical registration results are displayed in Figs. 1 and 2, respectively.

Fig. 1 manifests both ICP and SICP are effective in registering two 2D point sets with the same scale. However, SICP with unbounded scale works comparatively worse, for all points of 2D data shape converge to a small subset of 2D model shape. Hence, scale of SICP needs bounding.

In Fig. 2, ICP and SICP converge in a similar way at a fast speed while the curve of SICP with unbounded scale differs greatly and its RMS is much larger. In this experiment, the average performance time of each iterative step of these three algorithms is all less than 0.02 s, demonstrating our algorithm is efficient. In fact, at each iterative step, the average iterative numbers of the computation of scaling transformation are 2.22 and 2.44 for SICP and SICP with unbounded scale, respectively. Hence, although our algorithm cannot give an analytical solution for transformation parameters, it gets similar computation time as ICP, which is analyzed detailedly at the end of Section 3.2.

2. In the following experiment, some typical registrations are used to display SICP is able to complete registration with different scales while ICP cannot. First of all, the initial scale matrix and its constraint are also estimated from Eqs. (23) and (24). The compared results of these algorithms are given in Table 2.

In Table 2, $(\mathrm{RMS}_{ICP} - \mathrm{RMS}_{SICP})/\mathrm{RMS}_{ICP}$ is large, which suggests SICP is better than ICP in registration between two 2D shapes with different scales. Moreover, for beetle and turtle, whatever RMS of SICP with unbounded scale is less or more than that of SICP, its coordinate is close to 0, displaying the scale of SICP needs to be bounded. However, for bell, SICP with unbounded scale can get the same results as SICP, because SICP with unbounded scale just gets good initial parameters in this experiment. In Fact, the success or failure registration of SICP with unbounded scale depends on the estimation of the initial scaling transformation between two point sets. In some good cases, such as the shapes are simple or noises are rather little, the initial scaling transformation can be estimated quite well, so SICP with unbounded scale can get good registration results, but this is done difficultly in practice. As point sets are always undesirable, the initial parameters are always estimated inaccurate, which may cause the registration of SICP with unbounded scale to fail. Hence, the scale parameter of our algorithm is bounded in an interval, which can be tolerant of the inaccuracy of the estimation of the initial parameters. To see the compared results in a more intuitive way, one example of scaling registration results is portrayed in Figs. 3 and 4.

Fig. 3 displays SICP produces a more satisfying registration result than ICP, a proof to show SICP has good accuracy in 2D shape matching. In this figure, SICP with unbounded scale is worse than SICP, demonstrating the necessity of bounding the SICP's scale.

Table 3
Compared results of ICP and SICP on the Stanford database.

	Point sets	ICP	SICP	SICP with unbounded scale
Scale	Bunny	1.0000	*diag*(0.9786, 0.9919, 0.9561)	*diag*(0.5361, 0.4254, 0.3155)
	Dragon	1.0000	*diag*(1.0026, 1.0016, 0.9719)	*diag*(1.0026, 1.0016, 0.9719)
	Happy Buddha	1.0000	*diag*(0.9710, 0.9953, 0.9114)	*diag*(0.9712, 0.9953, 0.9112)
RMS ($\times 10^{-3}$)	Bunny	2.0217	1.9251	1.9935
	Dragon	1.8346	1.8141	1.8141
	Happy Buddha	2.2501	2.0949	2.0950

Fig. 4 shows the convergence of ICP, SICP and SICP with unbounded scale on two 2D shapes. In this experiment, all the algorithms converge monotonically and quite fast. The average steps of the transformation's computation are 1.42 and 2.16 for SICP and SICP with unbounded scale, respectively, and the average performance time of each iterative step of three algorithms is all about 0.01 s, proving that our algorithm is rapid.

5.2. Range data matching

In this part, our method is tested to prove its robustness and efficiency for registration of 3D range data. In the experiments we use the Stanford database, in which covariance matrices are applied to estimate the scale parameter and its constraint in the SICP algorithm. Moreover, to show the registration results in 3D space intuitively, view point(az,el) is used to set the viewing angle for a 3D plot. Azimuth, az, is a polar angle in the x–y plane, with the horizontal rotation about the z-axis as measured in degrees from the negative y-axis. Positive values of az indicate counterclockwise rotation of the viewpoint. Elevation, el, is the angle above (positive angle) or below (negative angle) the x–y plane.

Firstly, the Stanford database is used to compare ICP and SICP. Bunny (bun000 with 40,256 points and bun 045 with 40,097 points), Dragon (dragonStandRight_0 with 41,841 points and dragonStandRight_24 with 34,836 points) and Happy Buddha (happyStandRight_0 with 78,056 points and happyStandRight_24 with 75,582 points) are adopted. We estimate the initial scale matrix $\mathbf{S}_0$ and its constraint, and the compared results are displayed in Table 3, Figs. 5 and 6, respectively.

Displayed in Table 3, RMS of SICP is least, which proves that SICP can get higher precision. In addition, for Stanford Bunny, SICP with unbounded scale has small error, but its scale is also small,

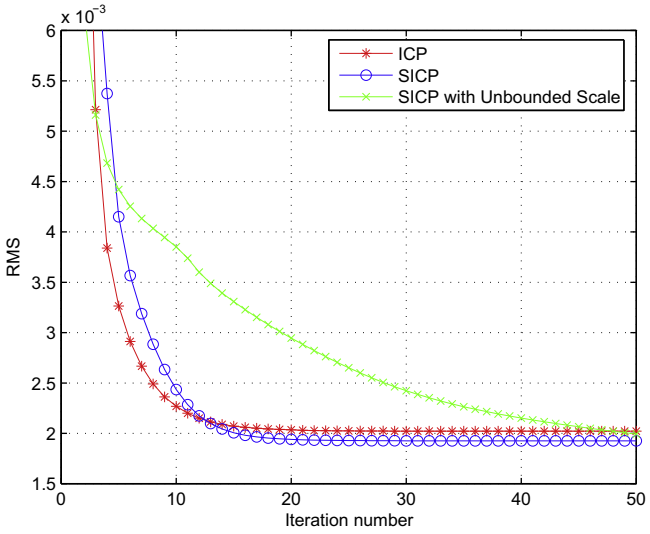

Fig. 6. The convergence of ICP, SICP and SICP with unbounded scale on the Stanford Bunny.

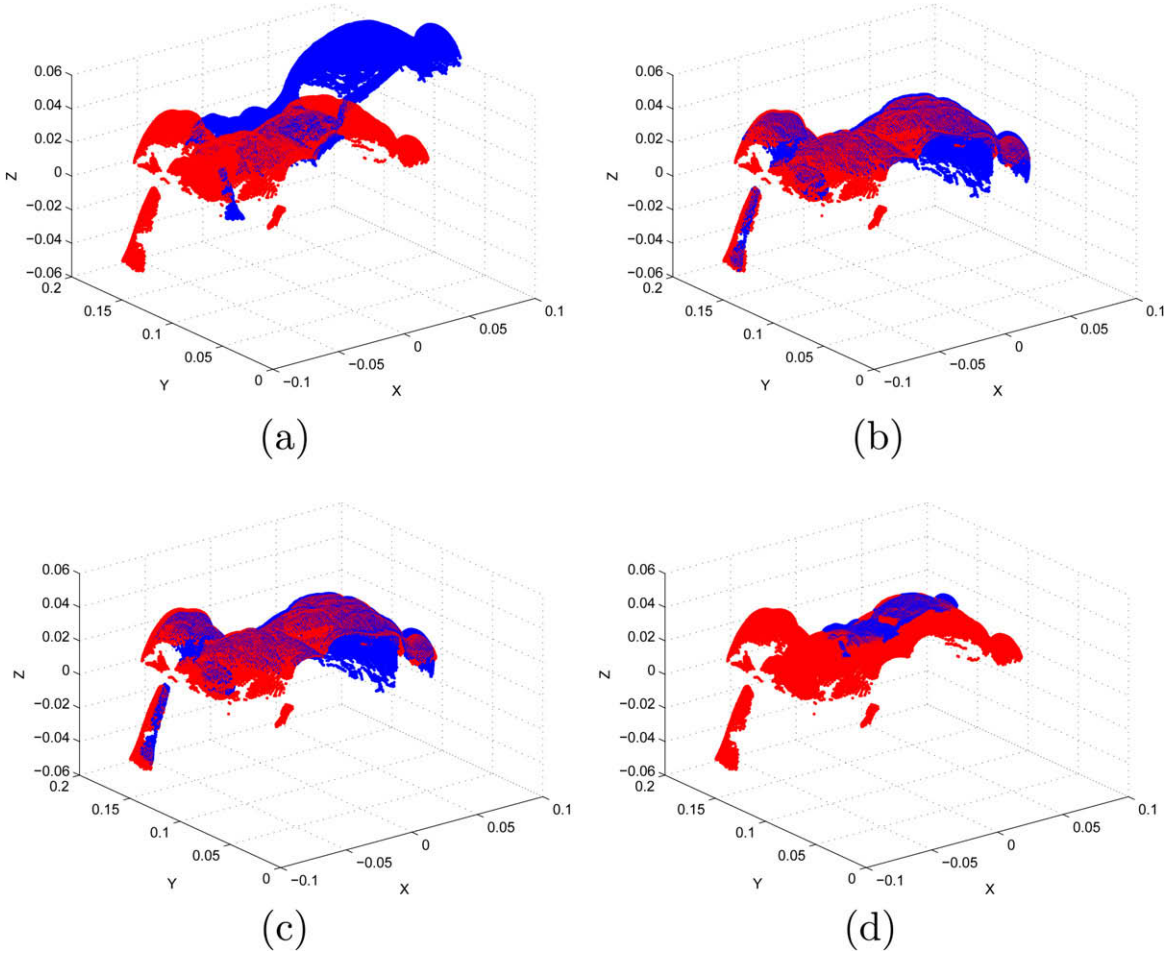

Fig. 5. Registration results on the Stanford Bunny. (a) Original data of Stanford Bunny. (b) Registration result of ICP. (c) Registration result of SICP. (d) Registration result of SICP with Unbounded Scale.

Table 4
Registration results of SICP on the Stanford Bunny with respect to the scale factor.

ρ_1	ρ_2	ρ_3	Scale S	RMS ($\times 10^{-3}$)	ST
0.01	0.01	0.01	*diag*(97.8707, 99.1987, 95.6056)	1.9251	*diag*(0.9787, 0.9920, 0.9561)
0.1	0.1	0.1	*diag*(9.7870, 9.9197, 9.5609)	1.9251	*diag*(0.9787, 0.9920, 0.9561)
0.5	0.5	0.5	*diag*(1.9575, 1.9838, 1.9124)	1.9251	*diag*(0.9787, 0.9919, 0.9562)
10	10	10	*diag*(0.0979, 0.0992, 0.0957)	1.9253	*diag*(0.9791, 0.9915, 0.9575)
100	100	100	*diag*(0.0098, 0.0099, 0.0096)	1.9254	*diag*(0.9793, 0.9913, 0.9582)

which denotes that SICP with unbounded scale fails in the registration of Stanford Bunny. This is demonstrated by Fig. 5 as well. Therefore, it is necessary to bound the scale of SICP. As is shown in Fig. 6, these three algorithms all converge monotonically. At each iterative step, the average steps of the transformation's computation are 4.54 and 4.24 for SICP and SICP with unbounded scale, respectively. Furthermore, the average performance time is 0.97 s, 1.00 s and 0.99 s for ICP, SICP and SICP with unbounded scale, respectively.

2. In the following experiments, the convergence of our algorithm is tested on Stanford Database with respect to the scale factor. We scale one range data to register the other range data and estimate the initial scale parameter and its constraint from Eqs. (23) and (24).

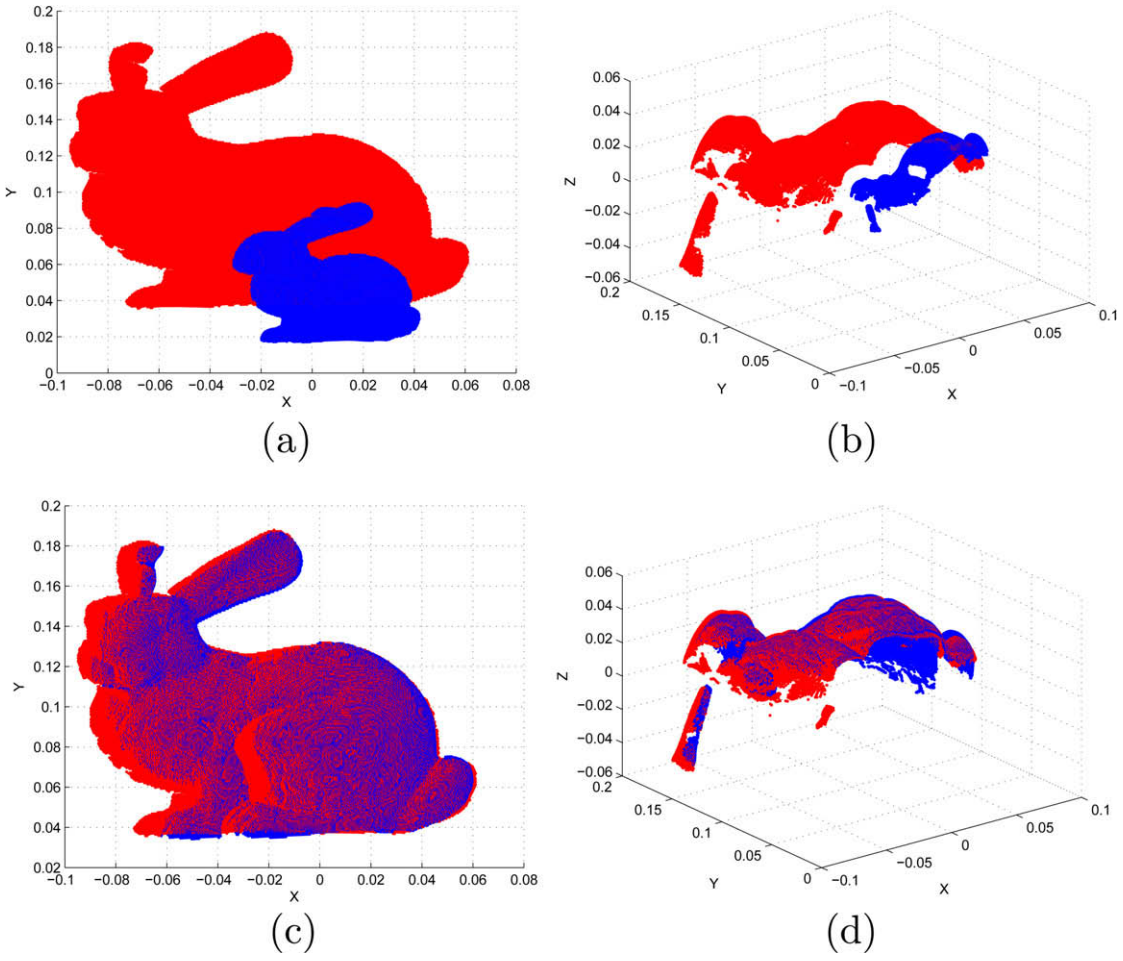

Fig. 7. Registration result of SICP on the Stanford Bunny when $T = diag(0.5, 0.5, 0.5)$. (a) Original range data with view point (0,90). (b) Original range data with view point (−37.5,30). (c) Registration result of SICP with view point (0,90). (d) Registration result of SICP with view point (−37.5,30).

Table 5
Registration results of SICP on the Stanford database with respect to the scale factor.

Range data	ρ_i	Scale S	RMS	ST
Dragon	0.01	*diag*(100.2684, 100.1593, 97.1604)	1.8141	*diag*(1.0027, 1.0016, 0.9716)
	0.5	*diag*(2.0053, 2.0031, 1.9435)	1.8141	*diag*(1.0026, 1.0016, 0.9717)
	100	*diag*(0.0100, 0.0100, 0.0097)	1.8141	*diag*(1.0024, 1.0016, 0.9723)
Happy Buddha	0.01	*diag*(97.0967, 99.5313, 91.1412)	2.0949	*diag*(0.9710, 0.9953, 0.9114)
	0.5	*diag*(1.9419, 1.9906, 1.8228)	2.0949	*diag*(0.9710, 0.9953, 0.9114)
	100	*diag*(0.0097, 0.0100, 0.0091)	2.0949	*diag*(0.9710, 0.9953, 0.9114)

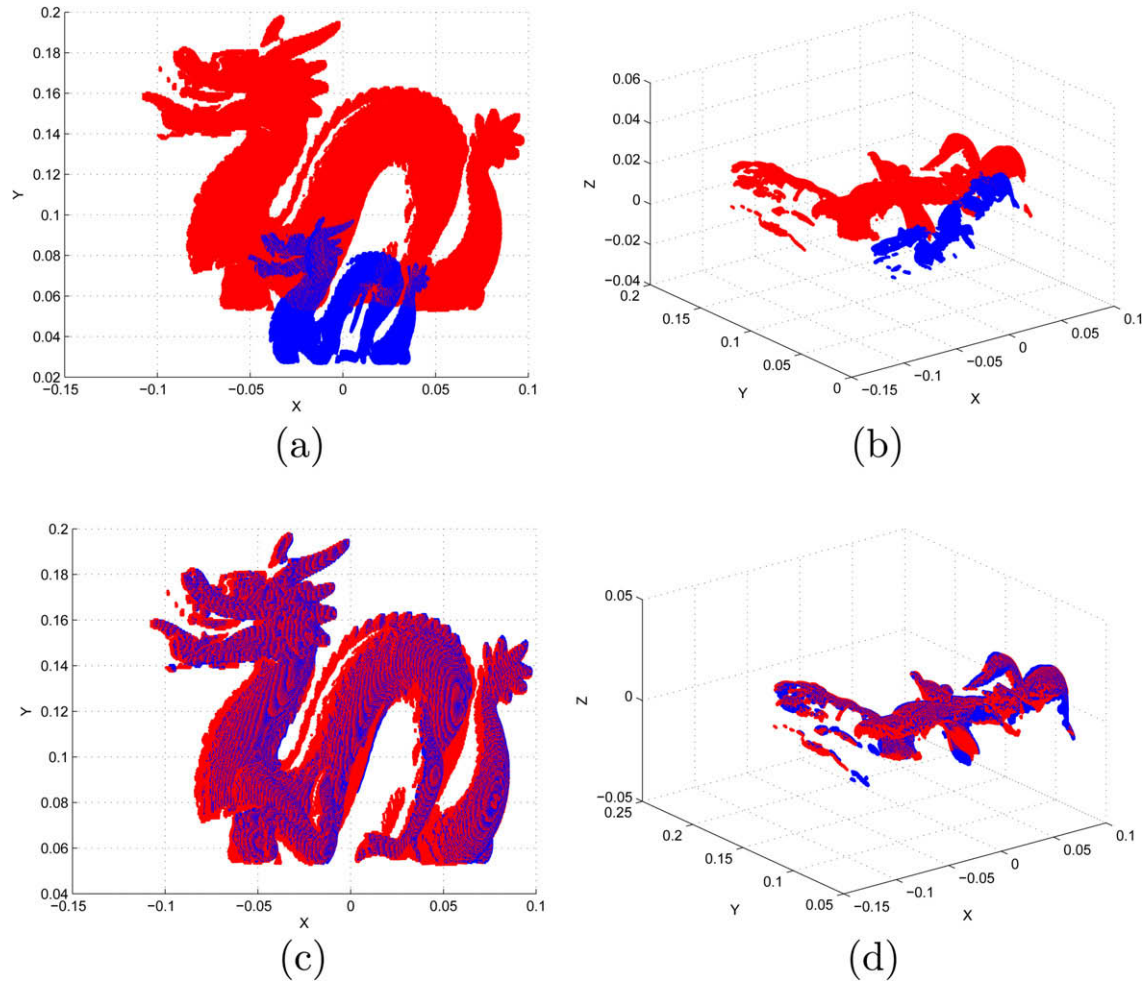

Fig. 8. Registration result of SICP on the Stanford Dragon when $T = diag(0.5, 0.5, 0.5)$. (a) Original range data with view point (0,90). (b) Original range data with view point (−37.5,30). (c) Registration result of SICP with view point (0,90). (d) Registration result of SICP with view point (−37.5,30).

We test our method on the Stanford Bunny first. We scale the bun045 by a scale factor $T = (diag(\rho_1, \rho_2, \rho_3))^{-1}$, and then use the zoom range data to register bun000. To demonstrate the stability of SICP's convergence, we let ρ_i vary greatly from 0.01 to 100. Some results are displayed in Table 4.

It is easy to see from Table 4 that whatever T is, RMS of SICP is similar. Nevertheless, the displayed scales appear quite different because the range data are scaled. To evaluate the scale factor of SICP in different experiments above, modified scale ST is introduced. The results reveal that the modified scales are similar. All of these above tell us that our method is robust in convergence with respect to the scale factor. Fig. 7 shows one fairly fine registration result of SICP.

We further test the algorithms on Dragon and Happy Buddha of Stanford Database. The procedure of these experiments is same as that on Stanford Bunny, and some results are shown on Table 5, from which, it is found that whatever T is, RMS and ST are similar. This proves again that our method is robust in convergence with respect to the scale factor. Figs. 8 and 9 show some good registration results of SICP.

6. Conclusion

This paper has advanced a novel approach for scaling registration between two m–D point sets in the way of incorporating a bounded scale matrix into the ICP algorithm. This algorithm uses a simple iterative algorithm with the SVD algorithm and the properties of parabola to compute the transformations at each iterative step. Due to its local convergence, the initial parameters need to be estimated by covariance matrices of two point sets. A series of compared experiments designed demonstrate our algorithm is more accurate and converges similarly in contrast to the standard ICP algorithm.

The main contributions of Scaling Iterative Closest Point (SICP) algorithm are delivered as follows:

1. It achieves similar accuracy as the standard ICP algorithm for the registration of point sets with the same scale, and it can handle the registration of point sets with different scales, while ICP cannot.
2. It has the similarly fast speed as the traditional ICP algorithm. Experiments demonstrate that they have similar performance time.
3. A novel technique has been proposed in our algorithm to obtain desired global minimum by estimating the initial parameters in the way of using eigenvalues and eigenvectors of covariance matrices of point sets.
4. Its scale parameter is required to be bounded. This is significant for converging stably in registration. Otherwise, points of a set converge to a small subset of the other.
5. It is a general framework for registration of m dimensions, and it can be easily used in conjunction with other algorithms as the ICP algorithm, for it is independent of shape representation and feature extraction.

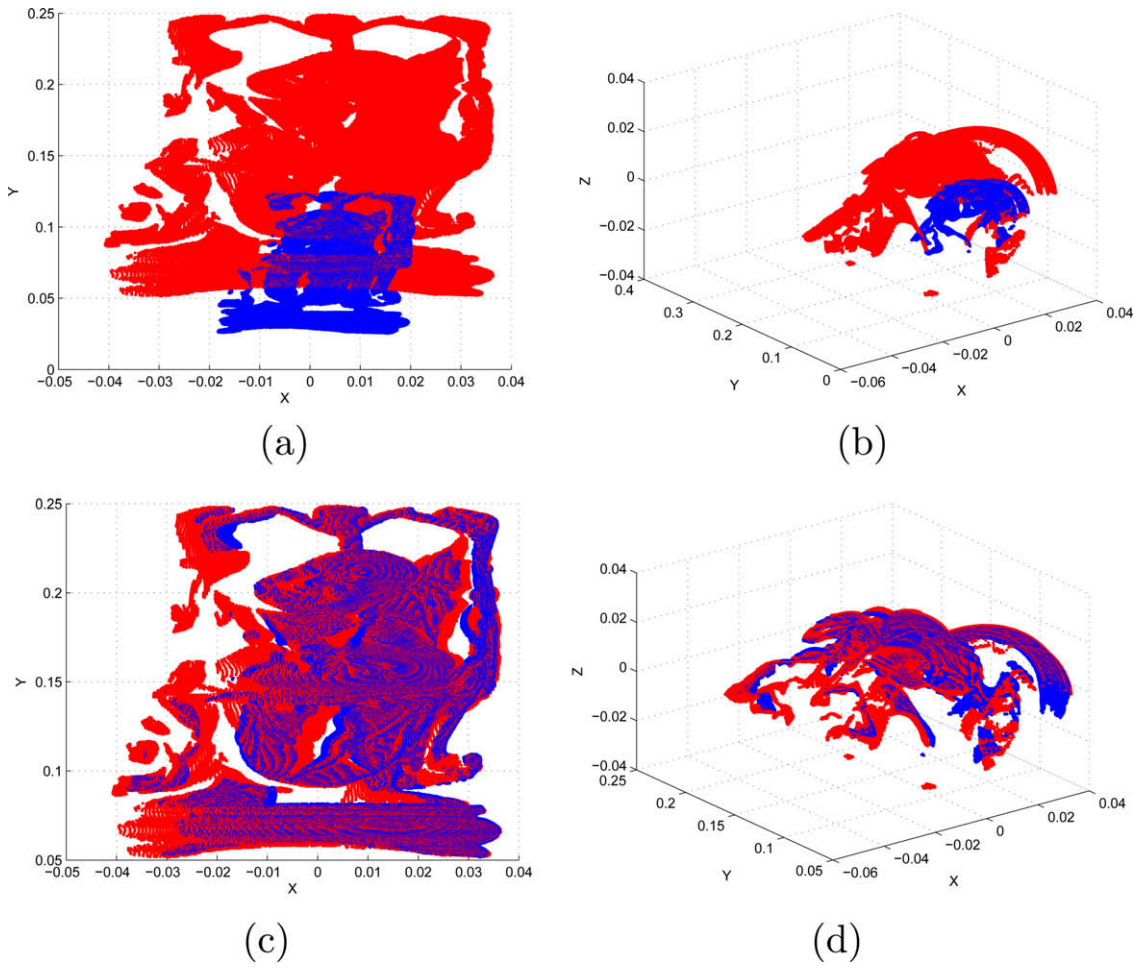

Fig. 9. Registration result of SICP on the Stanford Happy Buddha when $T = diag(0.5, 0.5, 0.5)$. (a) Original range data with view point (0,90). (b) Original range data with view point (−37.5,30). (c) Registration result of SICP with view point (0,90). (d) Registration result of SICP with view point (−37.5,30).

Though progress has been made in the work described heretofore, there is plenty of room for further numerous research issues. Our future work will introduce weights to improve the precision of registration between two partially overlapping point sets. Furthermore, actual testing of our algorithm also needs to be done for registration in practical use, such as 3D reconstruction of large ancient building and the like.

Acknowledgments

This work was supported by the National Basic Research Program of China under Grant No. 2007CB311005, and the National Natural Science Foundation of China Grant Nos. 60635050, 60875008, 90920008 and 90920301.

References

[1] P.J. Besl, N.D. McKay, A method for registration of 3D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256.
[2] Y. Chen, G. Medioni, Object modeling by registration of multiple range images, Image and Vision Computing 10 (3) (1992) 145–155.
[3] Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, International Journal of Computer Vision 13 (1) (1994) 119–152.
[4] X. Lu, A.K. Jain, D. Colbry, Matching 2.5D face scans to 3D models, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (1) (2006) 31–43.
[5] J. Minguez, L. Montesano, F. Lamiraux, Metric-based iterative closest point scan matching for sensor displacement estimation, IEEE Transactions on Robotics 22 (5) (2006) 1047–1054.
[6] K.W. Bowyer, K. Chang, P. Flynn, A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition, Computer Vision and Image Understanding 101 (1) (2006) 1–15.
[7] A. Almhdie, C. Leger, M. Deriche, R. Ledee, 3D registration using a new implementation of the ICP algorithm based on a comprehensive lookup matrix: application to medical imaging, Pattern Recognition Letters 28 (12) (2007) 1523–1533.
[8] A.W. Fitzgibbon, Robust registration of 2D and 3D point sets, in: Proceedings, British Machine Vision Conference, 2001, pp. 662–670.
[9] T. Jost, H. Hügli, A multi-resolution ICP with heuristic closest point search for fast and robust 3D registration of range images, in: Proceedings, Fourth International Conference on 3-D Digital Imaging and Modeling, 2003, pp. 427–433.
[10] B.U. Lee, C.M. Kim, R.H. Park, An orientation reliability matrix for the iterative closest point algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (10) (2000) 1205–1208.
[11] G.C. Sharp, S.W. Lee, D.K. Wehe, ICP registration using invariant features, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 90–102.
[12] S. Granger, X. Pennec, Multi-scale EM-ICP a fast and robust approach for surface registration, in: Proceedings, European Conference on Computer Vision, 2002, pp. 418–432.
[13] L. Silva, O.R.P. Bellon, K.L. Boyer, Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 762–776.
[14] H. Zha, M. Ikuta, T. Hasegawa, Registration of range images with different scanning resolutions, in: Proceedings, IEEE Int. Conf. Systems, Man, and Cybernetics, 2000, pp. 1495–1500.
[15] T. Zinßer, J. Schmidt, H. Niemann, Point set registration with integrated scale estimation, in: Proceedings, Int. Conf. on Pattern Recognition and Information Processing, 2005, pp, 116–119.
[16] S. Du, N. Zheng, S. Ying, Q. You, Y. Wu, An extension of the ICP algorithm considering scale factor, in: Proceedings, IEEE International Conference on Image Processing, 2007, pp, 193–196.

[17] C.B. Barber, D.P. Dobkin, H.T. Huhdanpaa, The Quickhull algorithm for convex hulls, ACM Transactions on Mathematical Software 22 (4) (1996) 469–483.
[18] K.S. Arun, T.S. Huang, Least-squares fitting of Two 3-D point sets, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (5) (1987) 698–700.
[19] B.K.P. Horn, Closed-form solution of absolute orientation using unit quaternions, Journal of the Optical Society of America, Series A 4 (4) (1987) 629–642.
[20] B.K.P. Horn, H.M. Hilden, S. Negahdaripour, Closed-form solution of absolute orientation using orthonormal quaternions, Journal of the Optical Society of America, Series A 5 (7) (1988) 1127–1135.
[21] M.W. Walker, L. Shao, R.A. Volz, Estimating 3-D location parameters using dual number quaternions, CVGIP: Image Understanding 54 (3) (1991) 358–367.
[22] A. Lorusso, D.W. Eggert, R.B. Fisher, A comparison of four algorithms for estimating 3d rigid transformations, in: Proceedings, British Machine Vision Conference, 1995, pp. 237–246.
[23] L.J. Latecki, R. Lakamper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2000, pp. 424–429.