Vol.2013-CVIM-187 No.1
2013/5/30

# Non-rigid Registration using Local Rigid Transformations for Shape Analysis

Kent Fujiwara[1,a)] Ko Nishino[2,b)] Jun Takamatsu[3,c)] Bo Zheng[1,d)] Katsushi Ikeuchi[1,e)]

***Abstract:*** With the advances in technology of sensing, it has become very easy to digitize various objects and capture detailed images and models. We propose non-rigid registration methods to utilize such data for shape analysis. We first introduce a novel dual-grid FFD (free-form deformation) framework which treats the source shape as a collection of local structures and match them to corresponding locations on the target shape. These local structures then guide the movement of FFD control points that determine the overall deformation. We then introduce a smoothness constraint and a weighting scheme based on distance from the surface to produce a smooth deformation grid and suppress movement in unnecessary regions. Finally, we extend the method to groupwise registration by estimating an "average" shape and apply the locally rigid globally non-rigid registration to each of the shapes in a group and transform them simultaneously.

## 1. Introduction

With the development of various sensors and the improvements to the sensing accuracy of these devices, it has now become easy to capture high quality images or models of various objects. One active field of research that requires these high quality images and models is shape analysis. Generally, shape analysis of actual objects is conducted by visual inspection or measurement of various characteristic parts. Methods such as destructive inspection, which involves actual physical interaction with the object, cannot be applied to valuable assets such as these artifacts. Using 3D data instead of real objects, a wider range of analysis methods can be applied to discover more about the subjects.

In this paper, we focus on the issue of comparing similar shapes and analyzing the relationship between them. We refer to this issue as *intra-class shape analysis*. We define it as a case of analyzing shapes that are classified into the same class, as opposed to *inter-class shape analysis*, which considers the issue of separating various objects into different classes.

One of the methods for intra-class shape analysis is solving for correspondences between objects through registration [5]. Registration is a method that transforms a source shape so that it would be aligned to a target shape. There are two main methods for registration: rigid and non-rigid registration. Rigid registration involves translation and rotation of the source shape. Iterative closest point (ICP) [1], [29] has been widely used to align point cloud data. Various strategies have been proposed to improve the original method [19], [22]. Non-rigid registration is a framework that requires a source shape to be deformed to a target shape to form correspondences between them. There are various methods of non-rigid registration that brings together images, point clouds, and volumes [7], [8], [10], [16], [17], [26], [31].

Our interest is in analyzing surface models of shapes from the same class, such as artifacts with similar shapes, or skulls and bones of the same animal species. The goal is to extract the similarities and differences within a group of similar shapes through non-rigid registration, and attempt to identify or describe each of the individuals in the group.

The first objective of this paper is to obtain a transformation between source and target objects that aligns the overall shape as well as the local prominent features. In order to conduct analysis on shapes that are classified into the same group, dense and meaningful correspondence between the shapes are required. In general, non-rigid registration methods focus on bringing object surfaces together as much as possible. However, the resulting transformations do not necessarily align the prominent features of these objects.

We propose methods that are able to align the overall surface of objects as well as their local features accurately and provide meaningful correspondences. In order to accomplish this goal, we introduce two methods that are based on the novel idea of "local rigidity and global deformability". In the non-rigid registration methods based on this idea, the source shape is considered as a group of local structures. These structures move to the most similar locations on the target shape. Then, these movements guide the overall deformation. By introducing the idea of local structures, the resulting deformation robustly aligns the overall shapes and the local features. This accurate correspondence, we claim, is essential to the analysis of similar objects that possess

[1] The University of Tokyo
[2] Drexel University
[3] Nara Institute of Science and Technology
[a)] kfuji@cvl.iis.u-tokyo.ac.jp
[b)] ko.nishino@drexel.edu
[c)] j-taka@is.naist.jp
[d)] zheng@cvl.iis.u-tokyo.ac.jp
[e)] ki@cvl.iis.u-tokyo.ac.jp

similar structures in the vicinity of each other.

The second objective is to obtain transformations between multiple similar objects that accurately align all the surfaces and the corresponding features. The same idea of "local rigidity and global deformability" will be introduced to achieve this goal. However, the two methods above cannot be directly applied to the scenario involving multiple images. This is due to the fact that the two methods are pairwise registration methods, which require a target shape to be designated. In the case of groupwise registration, which requires multiple shapes to be simultaneously aligned, this would force all the shapes to be aligned to the shape selected as the target, leading to a biased registration result.

We propose to overcome this by extending the proposed pairwise non-rigid registration method to handle multiple shape simultaneously. In this method, the target is estimated from all the shapes in the group to be registered. The estimated target, which would be the average of all the shapes, contains some "flat" regions in order to avoid the peaks from disrupting the average values. During the transformation of each shape, these flat regions are excluded from being included in the calculation of transformations, making the registration robust. By introducing this strategy, all the shapes as well as their local features are registered to the average position. This enables the simultaneous comparison of multiple shapes.

We conduct a number of experiments on 2D contour images and 3D surface models to demonstrate the effectiveness of the proposed methods. The results show that these methods achieve the objectives by aligning the shapes as well as their prominent features.

## 2. Locally Rigid Globally Non-rigid Surface Registration

The goal of non-rigid surface registration is to deform the surface of a source shape to match the surface of a target shape. We claim that an *ideal* surface registration should match characteristic features on the source shape to the corresponding features on the target shape. Prior work has attempted to accomplish this mainly by explicitly identifying corresponding features either by manual intervention or by feature extraction [11].

We search for this *ideal* registration by computing a shape-preserving deformation that brings the original shape into alignment with the target shape. We build on the key insight that the entire shape can be considered as a collection of local structures, each of which transforms rigidly to align with counterparts of the target shape and collectively deform the overall global structure.

We propose a novel dual-grid free-form deformation (FFD) representation to achieve the locally rigid, globally non-rigid registration. The secondary grid, which we call the sampling grid, is superimposed over the conventional FFD [23] control point grid that controls the overall deformation. The sampling grid is subdivided into sampling regions so that each control point is enclosed by one sampling region. The control point is translated based on the rigid transformation required to minimize the difference between the signed distance fields in the corresponding sampling region. Deformation is determined by the new positions of the control points

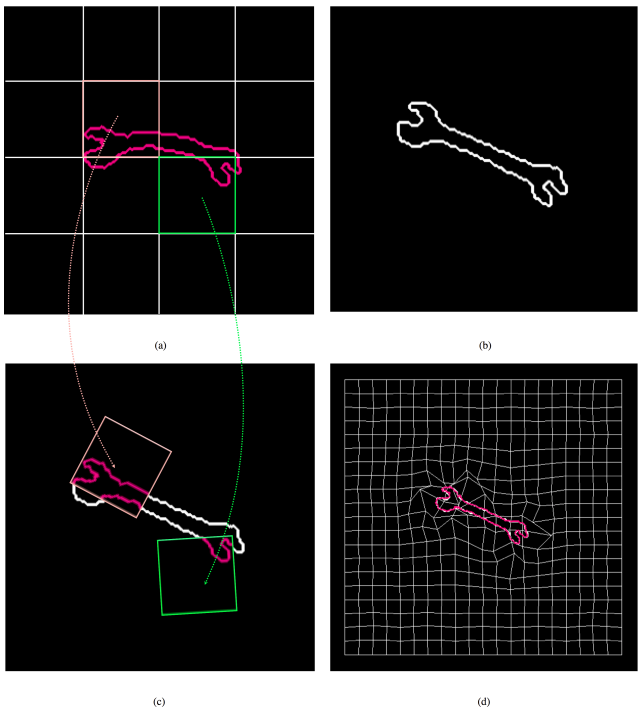



**Fig. 1** Our method deforms a source shape (a) to the target shape (b) by rigidly aligning local structures (c) that collectively form a free form deformation grid in a coarse-to-fine fashion (d).

### 2.1 Shape Representation

We first compute an implicit representation of the source and target shapes using the signed distance field (SDF). For the sake of simplicity, we explain this for 2D shapes but the same computation easily extends to 3D shapes.

Given a data shape $\mathbf{A}$, we consider an arbitrary point $\mathbf{x}$. From this arbitrary point, we search for the closest point on the data shape $\mathbf{A}$, which we denote as $\mathbf{a}_\mathbf{x} \in \mathbf{A}$. We assume that the surface normal of the shape is available at this point, which is expressed as $\mathbf{n}_{\mathbf{a}_\mathbf{x}}$. The signed distance $\phi_\mathbf{A}(\mathbf{x})$ of point $\mathbf{x}$ to the shape $\mathbf{A}$ is defined as the normal projected distance

$$\phi_\mathbf{A}(\mathbf{x}) = \mathbf{n}_{\mathbf{a}_\mathbf{x}}^T(\mathbf{x} - \mathbf{a}_\mathbf{x}) \,. \tag{1}$$

By calculating the signed distance at uniformly spread sampling points, we produce the SDF $\mathbf{\Phi}_\mathbf{A}$ around the shape $\mathbf{A}$.

### 2.2 Local Rigidity, Global Deformability

We propose a dual-grid FFD framework to achieve locally rigid and globally non-rigid registration. Figure 1 illustrates the overview of this framework. Here, we consider a registration process between a source shape $\mathbf{A}$ consisting of points $\mathbf{a} \in \mathbf{A}$ and a target shape $\mathbf{B}$ consisting of points $\mathbf{b} \in \mathbf{B}$. First, as in other methods using FFD, we prepare a grid that surrounds the source shape $\mathbf{A}$, which will be deformed in this process. This grid consists of FFD control points whose locations control the deformation of the field within. In Figure 2, the FFD grid is the gray grid.

In addition to this FFD grid, we add another grid, which we refer to as the sampling grid. We define each block of this sampling grid as sampling region $\mathbf{S}$ consisting of sampling points $\mathbf{s}$. In Figure 2, this grid is shown in red. The sampling grid is defined relative to the FFD grid so that a single FFD control point $\mathbf{P}_i$

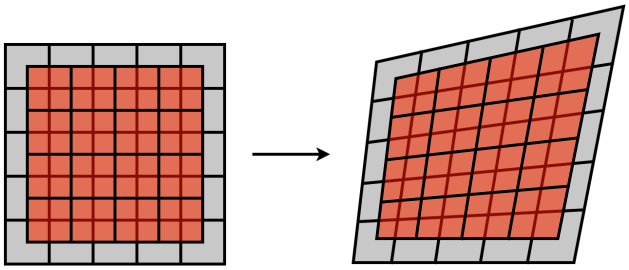


**Fig. 2** The sampling grid (red) is placed over the conventional FFD control point grid (gray). Each sampling region **S** transforms rigidly, moving the underlying control point **P** embedded within. The sampling grid is defined relative to the FFD grid so that the sampling grid completely covers the FFD grid without any overlap even after the FFD grid is deformed.

is immersed in a single sampling region $\mathbf{S}_{\mathbf{P}_i}$. This is achieved by defining the sampling region for each FFD control point by connecting the mid-points of the edges of the FFD grid connected to the control point of interest. This ensures that the sampling regions will be mutually exclusive and no space is sampled more than once. We place the same number of sampling points **s** and space them uniformly within each sampling region by defining them relative to the midpoints of the control points at the corners of the sampling region.

### 2.3 Local Rigid Registration

We solve for the optimal rigid transformation for each sampling region $\mathbf{S}_\mathbf{P}$ that minimizes the above error function. To register the two SDFs, we use the iterative registration method proposed by Lucas and Kanade [15]. The algorithm iteratively solves for the increments in the parameters $\Delta\mathbf{w}$ that minimizes the error between the SDFs and updates the estimated parameters $\mathbf{w}$ by solving for the error function defined as

$$E = \sum_{\mathbf{s}} \Big(\phi_\mathbf{B}\big(\mathbf{T}(\mathbf{s};\mathbf{w}+\Delta\mathbf{w})\big) - \phi_\mathbf{A}(\mathbf{s})\Big)^2 . \tag{2}$$

Taylor expansion can then be used to linearize this expression

$$E = \sum_{\mathbf{s}} \Big(\phi_\mathbf{B}\big(\mathbf{T}(\mathbf{s};\mathbf{w})\big) + \nabla\phi_\mathbf{B}\frac{\partial\mathbf{T}}{\partial\mathbf{w}}\Delta\mathbf{w} - \phi_\mathbf{A}(\mathbf{s})\Big)^2 , \tag{3}$$

where $\nabla\phi_\mathbf{B}$ represents the gradient of the signed distance. The partial derivative of the error function with respect to $\Delta\mathbf{w}$ is

$$\frac{\partial E}{\partial\Delta\mathbf{w}} = 2\sum_{\mathbf{s}}\left[\nabla\phi_\mathbf{B}\frac{\partial\mathbf{T}}{\partial\mathbf{w}}\right]^\mathsf{T}\left[\phi_\mathbf{B}\big(\mathbf{T}(\mathbf{s};\mathbf{w})\big) + \nabla\phi_\mathbf{B}\frac{\partial\mathbf{T}}{\partial\mathbf{w}}\Delta\mathbf{w} - \phi_\mathbf{A}(\mathbf{s})\right]. \tag{4}$$

This yields the additional values to the parameters $\Delta\mathbf{w}$

$$\Delta\mathbf{w} = \mathbf{H}^{-1}\sum_{\mathbf{s}}\left[\nabla\phi_\mathbf{B}\frac{\partial\mathbf{T}}{\partial\mathbf{w}}\right]^\mathsf{T}\left[\phi_\mathbf{A}(\mathbf{s}) - \phi_\mathbf{B}\big(\mathbf{T}(\mathbf{s};\mathbf{w})\big)\right], \tag{5}$$

where $\mathbf{H}$ is the Gauss-Newton approximation to the Hessian matrix which can also be written as

$$\mathbf{H} = \sum_{\mathbf{s}}\left[\nabla\phi_\mathbf{B}\frac{\partial\mathbf{T}}{\partial\mathbf{w}}\right]^\mathsf{T}\left[\nabla\phi_\mathbf{B}\frac{\partial\mathbf{T}}{\partial\mathbf{w}}\right]. \tag{6}$$

Deformation is defined by the coordinates of the control points on the FFD grid. The translation vector can therefore be applied

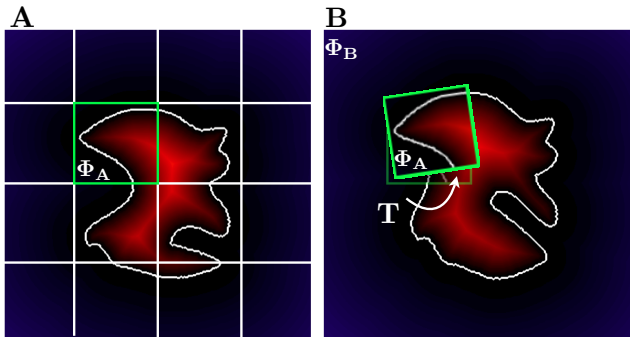



**Fig. 3** The difference between the signed distance field $\mathbf{\Phi}_\mathbf{A}$ of the sampling region of the source image **A** and the signed distance field $\mathbf{\Phi}_\mathbf{B}$ of the target image **B** is minimized by the rigid transformation **T** of the sampling region. This transformation guides the movement of the FFD control point, which determines the overall global deformation in the following step.

to move the control points, but the rotation, which rotates the sampling region around the control point, has no effect on the displacement of its center, that is, the control point itself.

To include the effect that a control point receives from the neighboring $n$ control points, we convert the rotation of neighboring control points $\mathbf{R}_i$ into translation $\mathbf{t}_\mathbf{R}$ of the control point at the center $\mathbf{P}$

$$\mathbf{t}_\mathbf{R} = \sum_i^n w_d\mathbf{R}_i(\mathbf{P}-\mathbf{P}_i), \tag{7}$$

where $w_d = \frac{w_i}{\sum_j^n w_j}$ is the weight based on the distance between the control points and $w_i = \frac{\sum_k^n d_k}{d_i}$ is the inverse of the normalized distance. $d_i$ represents the distance between the control point at the center $\mathbf{P}$ and the neighboring control point $\mathbf{P}_i$. The weight term is introduced based on the basic notion that the closer a sampling region is, the more the control point $\mathbf{P}$ should be influenced by it. This assigns more weight to the control points $\mathbf{P}_i$ that are closer to $\mathbf{P}$. The weights are normalized so that the total of the neighbor weights sum to one to avoid excessive effect from control points that are very close to each other.

The final translation would be

$$\mathbf{t}_\mathbf{f} = \mathbf{t}_\mathbf{s} + \mathbf{t}_\mathbf{R}, \tag{8}$$

where $\mathbf{t}_\mathbf{s}$ is the translation obtained from the transformation parameter $\mathbf{w}$.

### 2.4 Global Non-rigid Registration

We apply the rigid transformation obtained from each sampling region to the corresponding control point. Using these coordinates we move the FFD grid and apply the FFD deformation to the source shape. This process of computing the rigid transformation locally and applying it to the control points to deform globally is iterated until convergence.

Deformation of a point $\mathbf{a} = (x, y)$ using cubic B-spline FFD is computed by the weighted sum of the control points

$$\mathbf{F}(\mathbf{a}) = \sum_i\sum_j B_i(u)B_j(v)\mathbf{P}, \tag{9}$$

where $B$ denotes the B-spline basis function and $u$, $v$ represent the relative coordinates of $x$ and $y$ within the FFD grid. The global

deformation of the point $\mathbf{a}$ on the surface $\mathbf{A}$ is obtained by adding the translation $\mathbf{t_f}$ to the control point coordinates

$$\mathbf{F}(\mathbf{a}) = \sum_i \sum_j B_i(u) B_j(v) \Big( \mathbf{P} + \alpha_{\mathbf{S_P}} (\mathbf{t}_{\mathbf{f},i,j}) \Big).$$

When computing the translation of a control point from its sampling region, we weight the influence of the sampling region based on the area using the weight $\alpha$. This ensures that if the sampling region, which is defined relatively to the FFD grid, becomes smaller in the course of iterative computation of the deformation, its influence is lessened and eventually vanishes if it becomes degenerate. Therefore, $\alpha_{\mathbf{S_P},k} = \frac{S_{\mathbf{P},k}}{S_{\mathbf{P},0}}$ becomes smaller when the area of the sampling region $\mathbf{S}$ is smaller, and becomes larger when $\mathbf{S}$ is larger. Here, $S_{\mathbf{P},0}$ and $S_{\mathbf{P},k}$ are the area of sampling region around $\mathbf{P}$ at the initial stage and at the $k$-th iteration. This is normalized as $\alpha_{\mathbf{S_P},k} = \frac{\alpha_{\mathbf{S_P},k}}{\alpha_m}$ by dividing it by the largest value $\alpha_m$.

### 2.5 Experimental Results

We first evaluate the proposed method on 2D images using the silhouette data from the database provided by Sharvit *et al.* [24] and Bronstein *et al.* [4]. We extracted the contour data from these silhouette images and used them to test the registration accuracy. We compared the result of our method with the results of the distance field-based method by Huang *et al.* [11].

First, we obtained the results from Huang *et al.*'s method without the feature constraint. We then gave explicit feature correspondences to Huang *et al.*'s method and adjusted the parameters, such as the weight on the feature constraint and the bandwidth of SDFs to acquire the best possible registration results. The feature points that were provided are marked as blue dots on the source and the target image. These features are not given to our method, and are just used to quantitatively evaluate the registration accuracy. We also marked other features that were not given to any of the methods. These points, which are marked as pink dots, are also used to evaluate the accuracy of registration to gauge whether each method is capable of matching characteristic structures without explicit assignment of feature correspondences. For Huang *et al.*'s method with the feature constraint, this evaluates whether the registration computed from the given feature correspondences can propagate to align other features that were not specified. The corresponding features on the target surface are marked as white dots for visualization. In this section we present the registration results of "Device," "Fgen," "Misk," "Fish," and "Dude"in the database of Sharvit *et al.* [24] and "Pliers" and "Scissors" from the database of Bronstein *et al.* [4].

We used the same number of control points for the FFD grid in all of the methods. Huang *et al.*'s method uses a coarse-to-fine approach for registration and subdivides the FFD grid once the registration converges in the coarse stage. Therefore we applied the same approach to all the methods for comparison.

Some of the registration results are shown in Figs. 4 - 6. Figure 7 shows the residual error between all the feature points in various experiments including the 3 cases. In the case of Fig. 4, the previous methods come close to successful registration. However, the marked feature points did not match precisely. The result from our method is very accurate. Although these two shapes are quite different from each other, the assumption of local rigidity maintained the relative positions of the features and successfully matched them to the corresponding features.

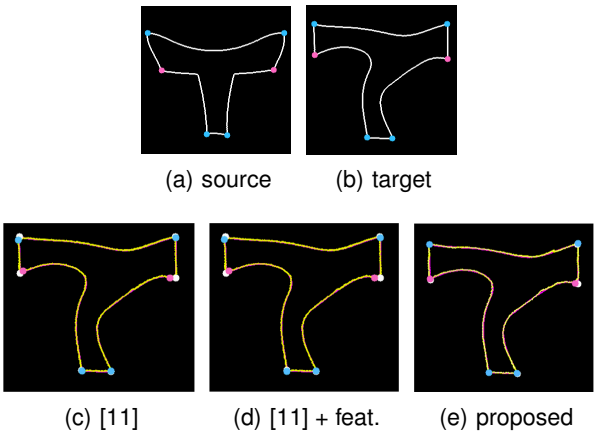



**Fig. 4** Experimental results of 2D contour data "device".

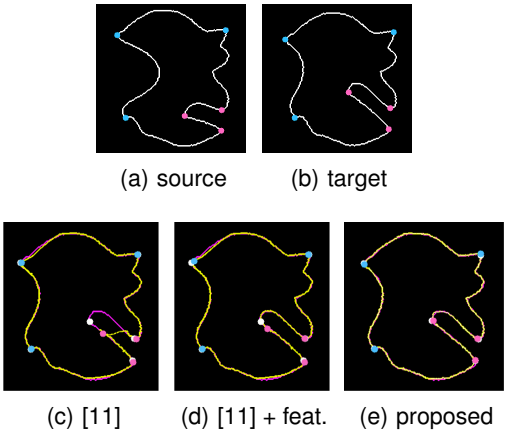



**Fig. 5** Experimental results of 2D contour data "misk".

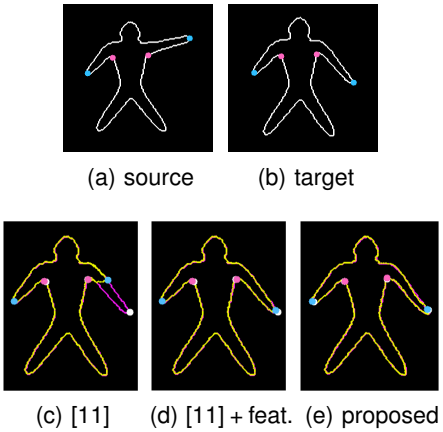



**Fig. 6** Experimental results of 2D contour data "dude".

In the case of Fig. 5, even though the previous methods succeeded in matching the outer surface and the blue features, the featureless version was stuck at a local minima. Huang *et al.*'s method using the features managed to match the inner surface, but the pink feature point inside the opening failed to match. Our method precisely matched this structure.

In Fig. 6, our method and the method by Huang *et al.* with the feature constraint performed well and matched the surfaces.

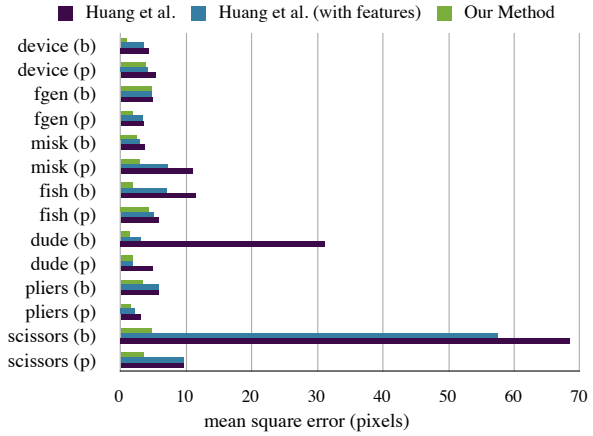



**Fig. 7** The mean square error between the blue feature points (b) and pink feature points (p) indicate that our method constantly outperforms other methods.

The featureless version of Huang *et al*. was stuck at a local minimum. The right arm of the image was difficult to register for the method by Huang *et al*. even with the given features, and required additional weight on the feature correspondences. However, this manual operation was still not enough to match the features on the right arm. This example shows that depending on the shape being registered, the method by Huang *et al*. requires manual parameter setting, as well as manual feature selection. Our method automatically achieves feature correspondence without any a priori correspondences by utilising the local structures.

### 2.6 Extension to 3D

Extending our method to handle 3D surface data is straightforward. We add the third coordinate $z$ to the vectors and matrices. The area of the sampling region used for calculating the weight $\alpha$ for global deformation is replaced with the volume of the 3D sampling volume. The rest of the method is the same as in 2D.

We have tested the 3D version of our method with synthetic 3D range data of a wave and a deformed version of the wave. Figure 8(a) shows the registration results. In Figure 8(b), we randomly deformed the "Stanford Bunny" [27] and registered the original model to the deformed model. We also processed 3D data taken from two separate chicken skulls to test our method on real data. Figure 8(c) is the result of registration. The experiments on synthetic and real data in 3D demonstrate that the method is effective in 3D as well.

Our concept of "align locally, deform globally" is based on the key idea that the shape along with the distance fields around it should maintain rigidity even in a non-rigid registration framework. Our method based on this key idea proved to be very effective in preserving local features and matching them to corresponding features on the target, which we believe is the most important aspect in non-rigid registration.

## 3. Non-rigid Registration with Local Rigid Transformations

The method in Section 2 treated all of the regions on the SDF equally. This caused the control points to move at locations where there is little or no effect on the actual deformation of the source

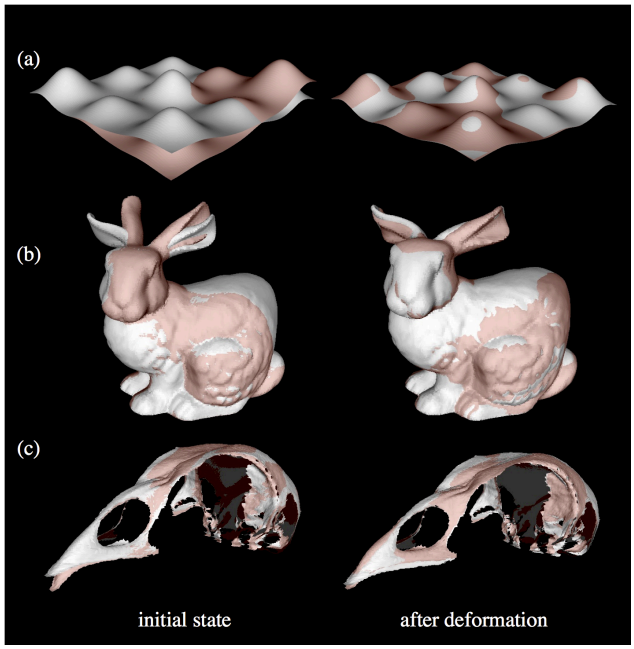



**Fig. 8** The method was applied to 3D synthetic data of waves (a), "Stanford Bunny" (b), and real data taken from actual skulls of chicken (c). The source image (red) was deformed to match the target image (white). Our method was successful at matching parts that were apart after initial rigid registration.

shape. The control points should ideally move in harmony with the neighboring control points to conduct robust alignment and obtain accurate correspondence.

We introduce a novel weighting scheme and a smoothness constraint based on the distance from the surface of shapes to achieve a flexible non-rigid registration, allowing more movement of control points when the difference between shapes is larger. The overview of our method is shown in Fig. 9.

In our locally-rigid globally non-rigid registration, we first divide the source shape and the space around it into sampling regions and then align each region to its optimal position in the entire target space. Registration of these sampling regions is analogous to template matching. Here, the sampling region with the local information of the source shape can be considered as a template that moves to a location in the target space where the difference is minimal. As depicted in Fig. 2, the sampling regions are defined to be the dual of the FFD control grid. The rigid transformation required to move each sampling region to the optimal position acts as a guide for the corresponding FFD control point encapsulated in each region. The rigid transformation of each sampling region indicates the translation of the corresponding control point required to achieve a global deformation that preserves the local structures as much as possible.

### 3.1 Formulation

In the proposed framework, we solve for the local rigid transformations that minimize the difference between the source SDF $\mathbf{\Phi_A}$ within the sampling regions and the entire target SDF $\mathbf{\Phi_B}$. The SDFs are defined in the same manner as in the previous section. Although the actual sampling grid and the FFD control grid are discrete grids, we first consider a special case where the sam-

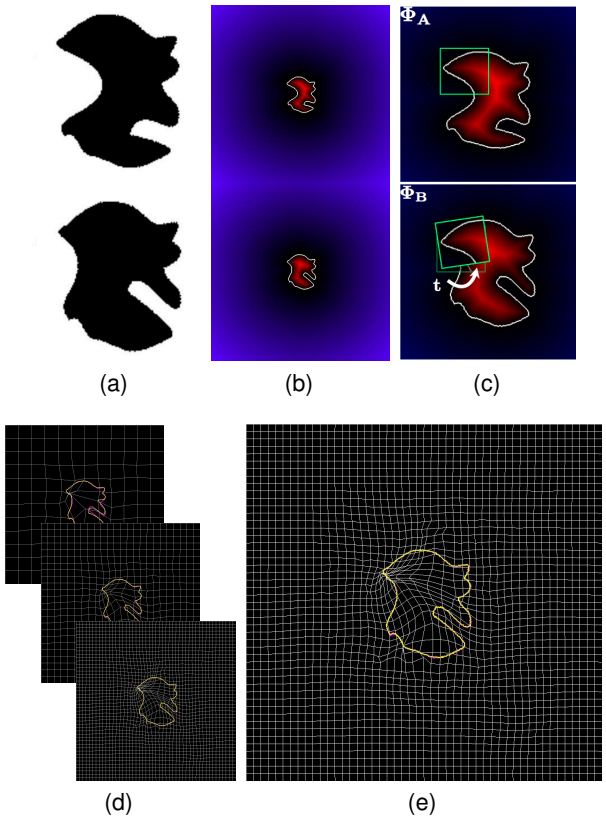



**Fig. 9** Overview of the proposed method. The input source and target shapes (a) are represented with the signed distance fields (SDF) (b). The sampling regions defined by the free-form deformation (FFD) control grid assumes rigid transformations that minimize the difference between the local SDFs and create a smooth deformation field (c). These transformations are applied to the corresponding control points to deform the source shape. This is conducted in a coarse-to-fine manner (d). By iterating the process in each of the scale levels, this method computes a smooth deformation field that accurately matches the overall shapes as well as the corresponding local characteristic structures.

pling regions are infinitesimally small and each region can be represented as a point $\mathbf{p} = (x, y)$ to simplify the notation. In this case, this infinitesimally small region is actually equivalent to the FFD control point $\mathbf{p} \in \mathbf{P}$, which is embedded at the center of each sampling region. In this context, rigid transformation of the sampling region can now be considered as translation. We will discretize the problem later to reflect the actual setting. The error function in this special case can be expressed as

$$E = E_e + E_s\,. \tag{10}$$

The error term $E_e$ is the difference between the source SDFs from the sampling regions and the entire target SDF. This is expressed as

$$E_e = \int\int \Big(\phi_{\mathbf{B}}\big(\mathbf{x} + \mathbf{t}(\mathbf{x})\big) - \phi_{\mathbf{A}}(\mathbf{x})\Big)^2 d\mathbf{x}\,, \tag{11}$$

where $\mathbf{t} = (u, v)$ is the translation in $x, y$ coordinates. $\phi_{\mathbf{A}}$ is the signed distance from a point to its closest point on shape $\mathbf{A}$. We introduce a smoothness constraint to this error function. As previously stated, this is to incorporate movements of neighboring sampling regions to make the entire FFD grid smooth. The smoothness term $E_s$ is represented as

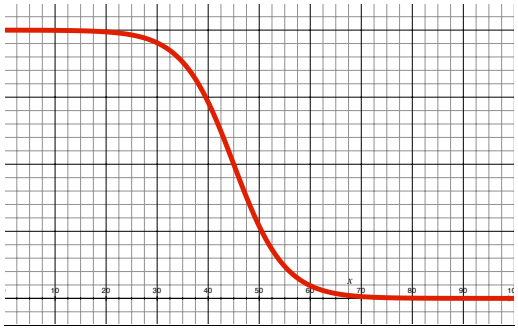


**Fig. 10** Visualization of the weight function $\alpha$. The weight (red) decreases as the distance from the surface (left boundary) increases, providing more weight to the smoothness constraint.

$$E_s = \lambda \int\int \Big(\mathbf{t}_x(\mathbf{x})^2 + \mathbf{t}_y(\mathbf{x})^2\Big) d\mathbf{x}\,, \tag{12}$$

where $\mathbf{t}_x(\mathbf{x}) = \frac{\partial}{\partial x}\mathbf{t}(\mathbf{x})$, and $\mathbf{t}_y(\mathbf{x}) = \frac{\partial}{\partial y}\mathbf{t}(\mathbf{x})$. $\lambda$ is the weight balancing the scale of the error term and the smoothness term. The goal is to find $\mathbf{t}$ for each sampling region that minimizes the error function.

### 3.2 Weight Based on Distance

In this formulation, all the locations are treated equally and would move to minimize the error between source and target SDFs. However, there are differences of importance among the regions. Regions away from the surface have large ambiguities in where they should move. Also, the regions far away from the surface only affect the control points that do not contribute at all to the deformation of the actual shape. Therefore, the control points closer to the surface of the shapes should have more effect on the overall registration than those away from the surface.

We take advantage of the pre-calculated distance values and introduce a weight $\alpha$ based on the distance from the surface. The weight is defined as a sigmoid curve, as shown in Fig. 10. The weighted version of the error term and the smoothness term are expressed as

$$\begin{aligned} E = \int\int & \alpha(\mathbf{x})\Big(\phi_{\mathbf{B}}\big(\mathbf{x} + \mathbf{t}(\mathbf{x})\big) - \phi_{\mathbf{A}}(\mathbf{x})\Big)^2 \\ & + \lambda\Big(1 - \alpha(\mathbf{x})\Big)\Big(\mathbf{t}_x(\mathbf{x})^2 + \mathbf{t}_y(\mathbf{x})^2\Big) d\mathbf{x}\,, \end{aligned} \tag{13}$$

where,$\alpha(\mathbf{x}) = \frac{1}{1+e^{n(m-k)}}$ and $m = \frac{1}{2}(|\phi_{\mathbf{A}}(\mathbf{x})| + |\phi_{\mathbf{B}}(\mathbf{x})|)$.

This means that if a control point is close to the surface, the error term receives more weight, and if it is far, the smoothness term becomes more prevalent. $n$ determines the steepness of the sigmoid function, and $k$ is the point where the value is halved. These two parameters are determined from the spacing of the initial FFD control grid.

The initial FFD control grid is set up so that the length of each side of the grid $l$ is 3 times as large as the largest side of the shapes. This is determined empirically to form a coarse grid in the initial stages of registration. The coarse grid ensures more space for each control point to move in, thus allowing more flexible registration.

In order for the control points to reflect the effect of this sigmoid function, there has to be at least 3 control points segmenting the curve of the sigmoid function from the sampling theorem.

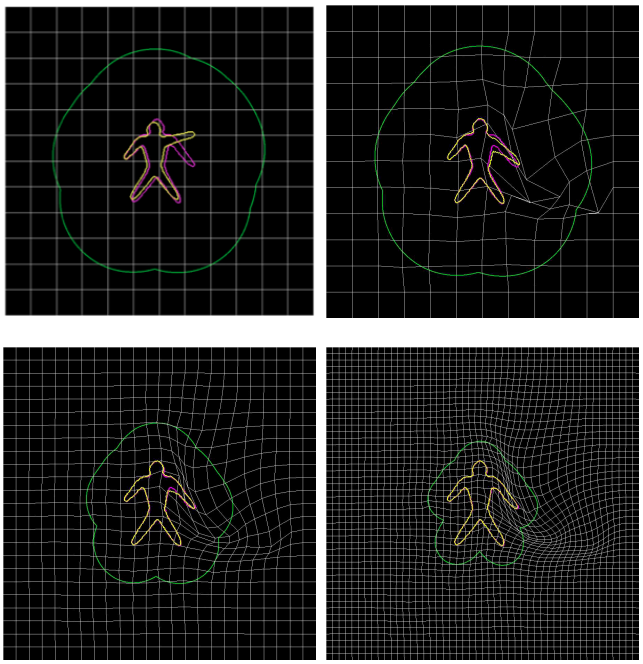


**Fig. 11** The hierarchical registration process. The process is iterated until convergence in the initial scale level. When the process converges, the grid is subdivided into a finer grid. The process is repeated in the finer scale levels until convergence. The sigmoid function, indicated by the region within the green line is recalculated based on the granularity of the FFD grid in each scale level.

We set the number of control points on each side of the initial grid to satisfy this condition. The space between the edge of the deformation field to the object surface is represented as $\frac{1}{2}(l-d)$, where $d$ is the diameter of the circle that encloses the shape, or the longest side of the input image. In this space, there has to be more than 4 control points to include the 3 control points and to also avoid moving the control points on the boundary of the FFD control grid. The intervals between control points should be at least $\frac{1}{8}(l-d)$. This means that on one side of the image $\frac{1}{2}l$, there has to be at least $\frac{1}{2}l \cdot \frac{8}{l-d} = \frac{4l}{l-d}$ control points. The same condition has to be fulfilled on the other side of the object, which leads to the overall number of control points $c = 2\lfloor \frac{4l}{l-d} \rfloor + 1$. The extra control point is added at the end to ensure the symmetry of the control grid. We calculate the gap between the control points $g = \frac{l}{c-1}$.

The parameter $k = \frac{3}{2}g$ of the sigmoid function is defined as the midpoint of the curve. The steepness of the curve is defined as $n = k^{-1}\log(199)$ so that the curve would be scaled properly and the weight should be as close to 1 as possible at the surface of the shape. The value of the weight smoothly decreases as the distance from the surface increases. The effect of the smoothness term is nullified in regions that are more than twice the value of $k$ away from the surface to avoid excessive movement of control points unrelated to the actual deformation of the surface.

### 3.3 Discretization

We conduct the optimization to estimate $\mathbf{t}$ using calculus of variations. The functional to consider is

$$F(\mathbf{x}, \alpha, \mathbf{t}, \mathbf{t}_x, \mathbf{t}_y) = \alpha(\mathbf{x})\Big(\phi_{\mathbf{B}}\Big(\mathbf{x} + \mathbf{t}(\mathbf{x})\Big) - \phi_{\mathbf{A}}(\mathbf{x})\Big)^2 + \lambda\Big(1 - \alpha(\mathbf{x})\Big)\Big(\mathbf{t}_x(\mathbf{x})^2 + \mathbf{t}_y(\mathbf{x})^2\Big). \quad (14)$$

We considered a special case where sampling regions were infinitesimally small and were equivalent to points to make the problem easier. However, this does not reflect the actual setting where the sampling grid and the FFD control grid are discrete, and each of the control points is encapsulated in a sampling region. We, therefore, discretize the problem and solve for $\mathbf{t}$ of each sampling region, which also determines the movement of the corresponding control point $\mathbf{p}$. Assuming that the boundary condition is fulfilled, which in this case would be $\frac{\partial}{\partial \mathbf{t}_x}F = \frac{\partial}{\partial \mathbf{t}_y}F = 0$, the Euler-Lagrange equation, by abuse of notation, is

$$\frac{1}{n}\alpha(\mathbf{p})\sum_i^n \Big(\phi_{\mathbf{B}}(\mathbf{p} + \mathbf{s}_i + \mathbf{t}(\mathbf{p})) - \phi_{\mathbf{A}}(\mathbf{p})\Big)\frac{\partial}{\partial \mathbf{t}}\phi_{\mathbf{B}}(\mathbf{p} + \mathbf{s}_i + \mathbf{t}(\mathbf{p})) + \lambda\Big(\alpha_x(\mathbf{p})\mathbf{t}_x(\mathbf{p}) + \alpha_y(\mathbf{p})\mathbf{t}_y(\mathbf{p}) - (1 - \alpha(\mathbf{p}))\Delta\mathbf{t}(\mathbf{p})\Big) = 0\,, \quad (15)$$

where $\alpha_x(\mathbf{x}) = \frac{\partial}{\partial x}\alpha(\mathbf{x})$, $\alpha_y(\mathbf{x}) = \frac{\partial}{\partial y}\alpha(\mathbf{x})$, and $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ denotes the Laplace operator. $\mathbf{s}_i = (s_i^x, s_i^y)$ is the sampling point in the sampling region surrounding the control point. The sampling points are defined in the local coordinate system with the origin at the corresponding control point. The translation can be computed numerically by approximating the Laplacian as $\Delta\mathbf{t}(\mathbf{x}) = \bar{\mathbf{t}}(\mathbf{x}) - \mathbf{t}(\mathbf{x})$, where $\bar{\mathbf{t}}(\mathbf{x})$ is the weighted average of the neighbor translations. The movements of the neighboring control points from the previous iteration are used to solve the above.

Deformation of a point $\mathbf{a} = (a_x, a_y)$ on the surface $\mathbf{A}$ using cubic B-spline FFD is computed by adding the translation $\mathbf{t}_{ij} = (u_{ij}, v_{ij})$ to the coordinates of the corresponding control point:

$$\mathbf{T}(a_x, a_y) = \sum_i \sum_j B_i(s)B_j(t)\Big(\mathbf{p}_{ij} + \mathbf{t}_{ij}\Big), \quad (16)$$

where $B$ denotes the B-spline basis function and $s$, $t$ represent the relative coordinates of $a_x$ and $a_y$ within the FFD grid. After deformation, the sampling grid is redefined according to the transformed FFD grid. The calculation of local transformation and global deformation is conducted iteratively until convergence.

### 3.4 Hierarchical Approach

We iterate the deformation using the initial grid until the control points cease to move. We then subdivide the FFD grid into a finer grid using the method proposed by Lee *et al*. [13]. Fig. 11 shows the registration results for each level for an example 2D shape pair .

This coarse-to-fine strategy is essential to our framework. The division size of the FFD grid determines the size of the sampling regions. The sampling regions determine the local regions of the source shape that should move rigidly. Therefore, the FFD grid size controls the scale of local structures of the source shape that rigidly align to the target shape to collectively achieve global registration.

In order to align large-scale characteristic structures first, and then match finer-scale local structures, we progressively subdivide the grid so that every time the registration converges, a new round of iteration with finer sampling regions is started.

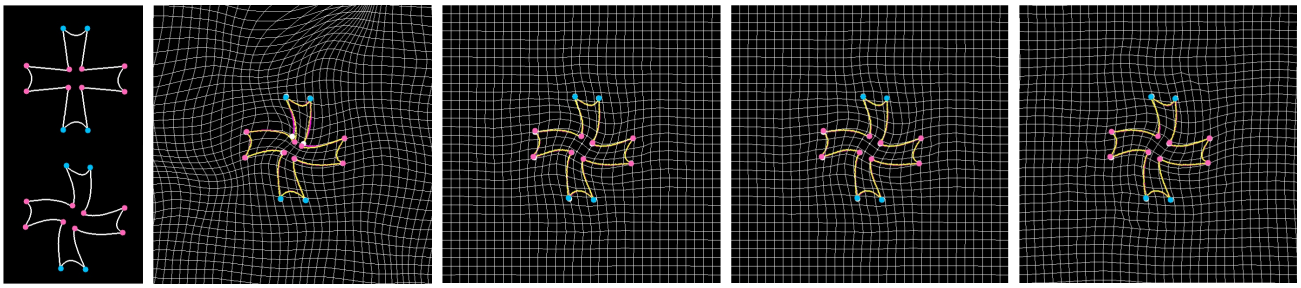



**Fig. 12** Experimental results of 2D contour data “device 2”.

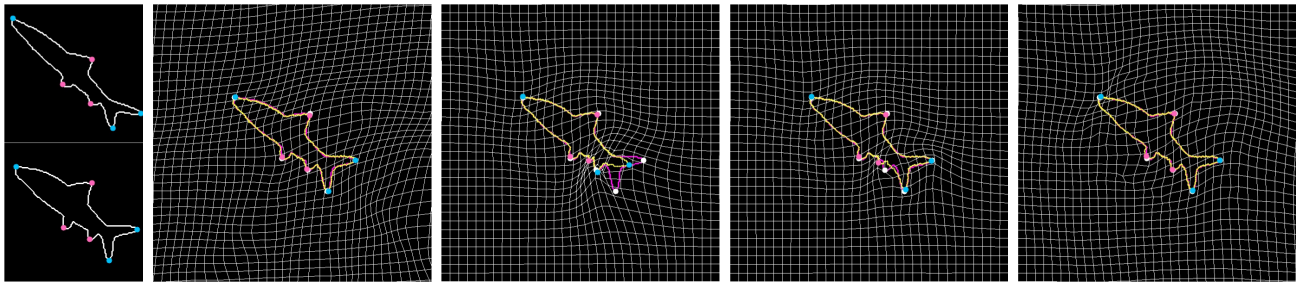



**Fig. 13** Experimental results of 2D contour data “fish”.

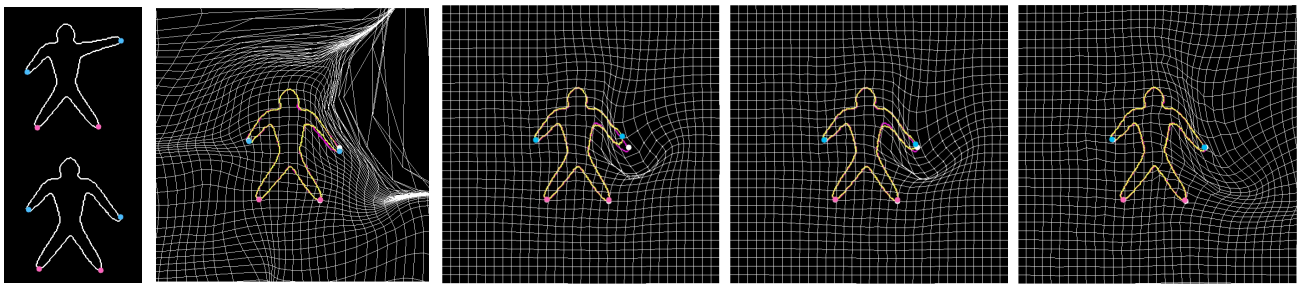



**Fig. 14** Experimental results of 2D contour data “dude”.

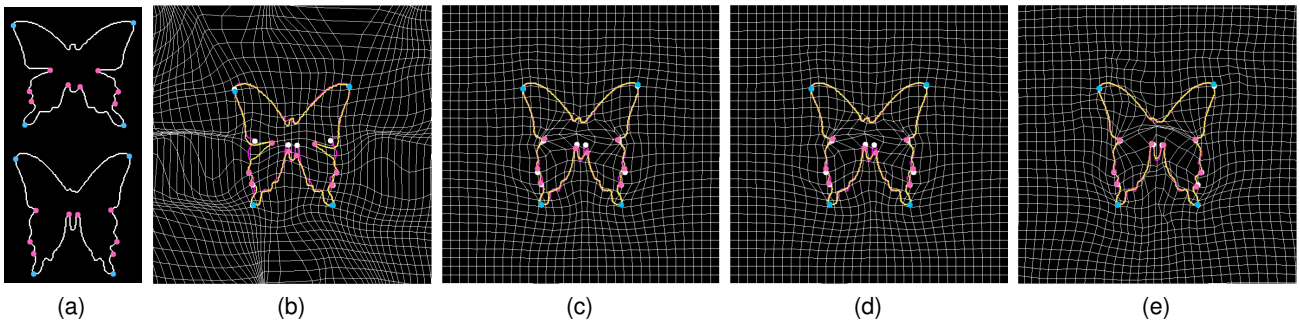



**Fig. 15** Experimental results of 2D contour data “butterfly”. The input source image ((a) top) is deformed to the target image ((a) bottom) by 5 different methods: (b) Likar and Pernuš, (c) Huang *et al*, (d) Huang *et al*. using blue points as given features, and (e) the proposed method.

### 3.5 Extension to 3D data

The proposed method can easily be extended to align 3D surface data. The sampling regions that were introduced in the 2D case will be converted into sampling volumes. A sampling volume would be defined as a 3D space bound by the midpoints of the corresponding control point and its neighbors.

The 3D algorithm also requires the third coordinate $z$ and its translation $w$ to be added to the Euler-Lagrange equation. The same weighting scheme can be applied to the 3D case and no other modifications to the original method is necessary.

### 3.6 Experiments

We first evaluate the effectiveness of the proposed method using 2D contour shapes obtained from silhouette images. We use the data from the silhouette image database provided by Sharvit *et al*. [24], MPEG-7 CE-Shape-1 database [12], and Bronstein *et*

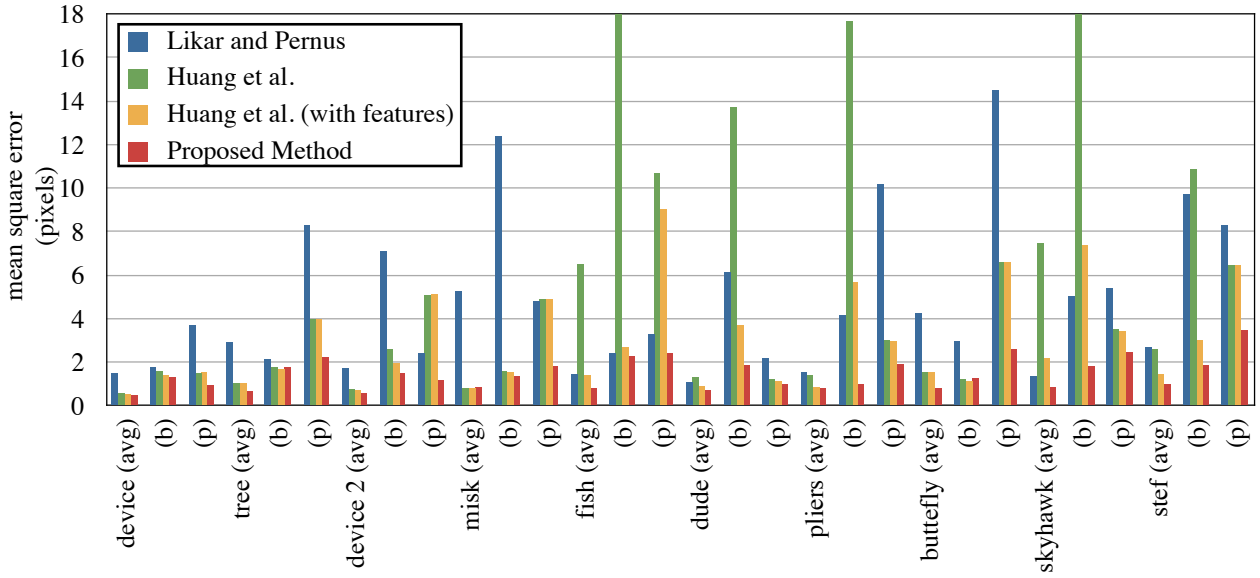



**Fig. 16** Overall accuracy and distance between the corresponding features. (avg) is the average distance between the sampled points on the source shape and the closest point on the target. (b) represents the difference between the corresponding blue points, and (p) the difference between the corresponding pink points.

*al*. [4]. We compared the results of the proposed method with those by Likar and Pernuš [14] that employs a subdivision strategy, and Huang *et al*. [11] that uses the SDF and FFD. It is also possible to provide explicit feature correspondence to the method by Huang *et al*. as an additional constraint, which we compare our method with as well.

#### 3.6.1 Overall Registration Accuracy and Feature Correspondence

To evaluate the effectiveness of the locally rigid but globally non-rigid registration, we compared the accuracy of surface registration and characteristic feature correspondence.

As for the method by Likar and Pernuš, we treated the SDFs as intensity images, then calculated the mutual information and conducted the registration process. As for the method by Huang *et al*. we registered the images without the feature constraints and compared the results. Then, we assigned correspondences to the features and used this information for the feature constraint in the method by Huang *et al*. The results of these three methods were compared with the results from the proposed method. The source code for the method by Likar and Pernuš was not available, and was implemented based on the information from the paper. However, the restriction based on distinctness and similarity was removed in this experiment because of some unstable behavior. Instead, the deformation process was repeated in each layer. This was not mentioned in the original paper, but made the method more robust.

The overall registration accuracy in each method was measured by calculating the average distance between the sampled points on the deformed source surface and their closest points on the target surface. Every 10th point on the source shape was used for this calculation.

The accuracy of the feature correspondence was measured by calculating the average distance between the corresponding blue and pink points. The blue points are feature points whose correspondence information was given to the method by Huang *et al*. for the feature constraint. These feature correspondences are not given to any other methods and are used solely for quantitative evaluation. The pink points are feature points whose correspondence information was not given to any of the methods. These points are used to measure how well each method can handle regions whose correspondence is unknown. The points on the target surface is marked as white points for visualization of matching accuracy.

The experiments were conducted using “device”, “tree”, “device2”, “misk” and “fish”, whose corresponding structures are relatively close to each other, and “dude”, “pliers”, “butterfly”, “skyhawk” and “stef”, which involve parts that are relatively far. The images of “misk”, “fish”, “dude”, “butterfly”, “skyhawk” and “stef” are from the database of Sharvit *et al*. [24], “tree”, “device” and “device2” are from MPEG-7 CE-Shape-1 database [12], and “pliers” images are provided by Bronstein *et al*. [4].

We set up the same FFD control grid for each of the methods. The coarse-to-fine approach was adopted for all of the methods. The FFD grid was subdivided exactly twice. The method by Likar and Pernuš uses thin-plate splines for deformation and does not have the same grid to represent the deformation of the field. To visualize how much the space had deformed in their method, the grid with the same number of points as the other methods in the final stage of subdivision was placed and deformed. This grid does not have any effect on the registration. The number of bins for the calculation of mutual information in the method by Likar and Pernuš was set to 400, which is the average distance from

the surface of the object to the edge of the deformation field, but was increased up to 600 when the registration accuracy was poor. The distance threshold that limits excessive movement of each block was set to half the side length of rigid structures. The subdivision of images was conducted 4 times, and 30 iterations were conducted in each layer.

The parameters in the method by Huang *et al*. controlling the smoothness, the weight on feature correspondence, and the bandwidth of SDF, were set to 5.0, 3.0 and 30.0, respectively. This setting performed well in most of the cases. However, this did not yield the best results in all of the cases. To make the comparison fair and obtain the best possible results, the parameters were altered within a reasonable range (maximum of 10 times the initial value) depending on each case, especially when the difference between shapes was larger.

Figs. 12 - 15 are some of the registration results. Fig. 16 shows the overall registration error and the distance between corresponding feature points.

The first two rows show the registration results from cases where the source and target shapes are relatively similar. The quantitative evaluation in Fig. 16 shows that all the methods are relatively successful at aligning the surfaces together. However, there are some larger errors in some cases, such as "fish" deformed by the method of Huang *et al*. The feature constraint in the method by Huang *et al*. was effective in bringing the blue feature points close to the corresponding points on the target but the method failed in regions where correspondence was not given, which can be observed from the residual error of the pink feature points.

The bottom two rows are the registration results from cases where the difference between the source and target shapes is larger. Here, the overall accuracy of registration become lower in many cases. The method by Likar and Pernuš fails to align many parts of "butterfly". The method by Huang *et al*. without any feature constraint fails to align the blue points as well as the pink points. The feature constraint introduced to the method by Huang *et al*. reduces this error to some extent. The blue points in this case are much closer to the corresponding points. However, this constraint does not affect the regions near the pink points where correspondence information is unavailable.

The results from our method, even when the difference between shapes is relatively large, are very accurate compared to the other methods. The method was able to find the similar structures in the vicinity and perform a desirable deformation.

These results show that our method is capable of deforming the source shape to the target shape as well as aligning characteristic features to the corresponding part on the target shape without any intervention.

### 3.6.2 Grid Smoothness and Flexibility

We further evaluate our proposed method by analyzing the smoothness of the deformation field obtained from each registration method. We recorded the FFD grid of each method after convergence. For each control point of the FFD grid, we calculated the average distance to the neighboring control points.

The method by Likar and Pernuš was removed from this comparison because it does not use FFD for deformation and the grid

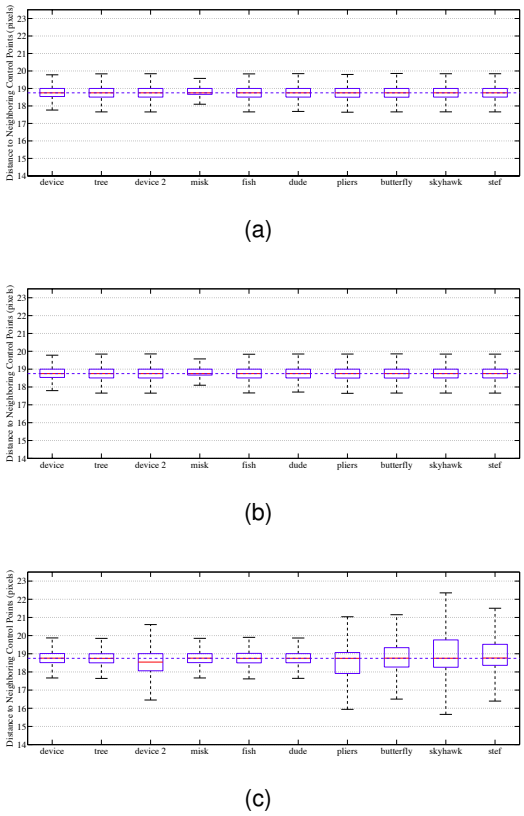



**Fig. 17** Distribution of the average spacing between the neighboring control points. The whiskers provide 99.7% ($3\sigma$) coverage. (a) Results from method by Huang *et al*. (b) Results from method by Huang *et al*. using the blue points for feature constraint. (c) Results from the proposed method. Blue dotted line represents the original spacing between the control points when no deformation takes place.

was placed for visualization. However, as observed in Figs. 12 - 15, the grid becomes very distorted when the difference between source and target shapes becomes larger. The regions away from the surface are also deformed because the method originally targets intensity images and do not constrain the movements of the outer regions.

Fig. 17 shows the boxplots of the distributions of the average distance between the neighboring control points. The blue horizontal line shows the distance between the control points of the undeformed FFD grid.

The first row shows the results from the method by Huang *et al*. without any feature correspondence. The second row is the results from the method by Huang *et al*. using the blue points as the feature constraint. The results from our method are shown in the last row.

In cases where shapes are relatively similar, the boxplots from the three methods are very similar. This indicates that our method was able to register the shapes more accurately than the previous methods with the similar amount of grid movement as the previous methods.

When the difference between the shapes becomes larger, the variance of the distributions from our method becomes larger while the results from the other methods remained similar to the other distributions. The hierarchical weighting scheme of our method moved more control points in cases where difference between shapes is large. In the initial stages, the outer control points

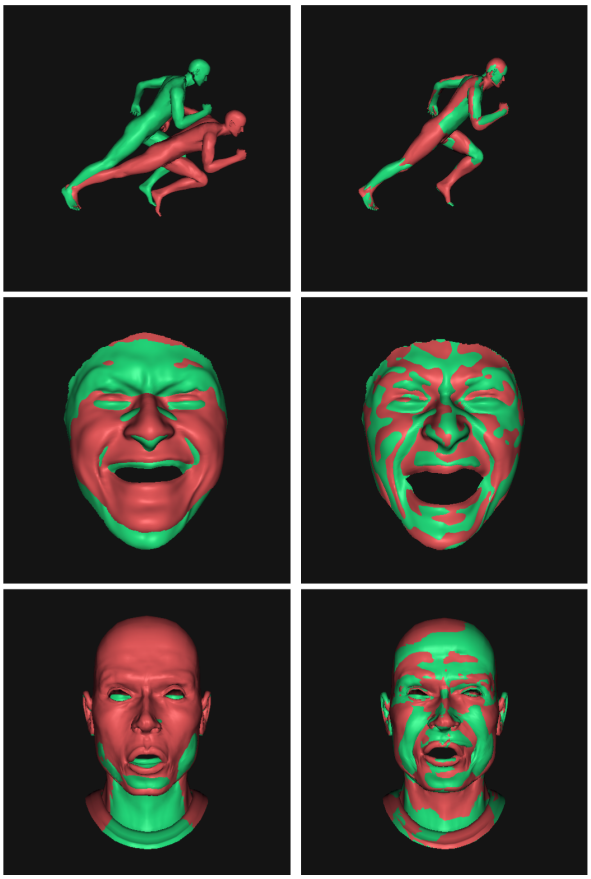



**Fig. 18** Experimental results of 3D range data of "michael", "face", and "head" (a) Initial state. (b) After deformation. Source and target are shown in red and green, respectively.

within the range of the sigmoid weight function were allowed to move to some extent in order to handle larger differences and conduct a precise alignment of the characteristic features. On the contrary, the method by Huang *et al.* sets the bandwidth of the SDF to be quite narrow (30 pixels in most cases). The design of the previous method led to less flexibility and left behind unaligned regions. This is fatal to analysis of highly detailed images, where accurate correspondence is essential.

The flexibility of our method maintains the initial grid as much as possible in cases where little deformation is required, and allows more grid movement in cases where difference between shapes is large. As a result, the overall shape and the local structures are accurately aligned.

#### 3.6.3 3D Surface Data

We also applied our method to 3D surface data. As previously described, the proposed method can easily be extended by adding another dimension to the 2D formulation and replacing sampling regions with sampling volumes.

In this experiment, we used "michael" data from the database of Bronstein *et al.* [3], and "head" and "face" data from the database of Sumner and Popovic [25].

Fig. 18 show the 3D registration results. The method also handled relatively large positional difference as is seen from the case of the man running. In the case of the head data, the method successfully brought together the corresponding parts, such as the eyes, the nose and the mouse and aligned two different expressions. Accurate results were also obtained from the face data, where a laughing face was aligned to a furious face.

From these results, we can easily observe the effectiveness of the proposed registration method in 3D. Extension to 3D surface data only requires a simple process of providing the $z$ coordinates to all the calculations and replacing the sampling regions with sampling volumes.

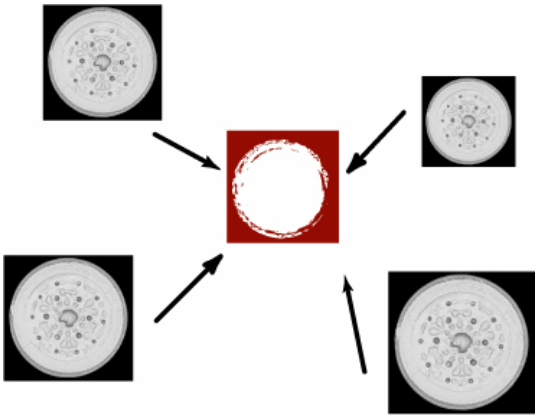


**Fig. 19** Groupwise registration. All the source shapes are aligned to a certain unknown shape (red). This shape generally represents the "average" shape of all the data in the group.

## 4. Groupwise Non-rigid Registration with Local Rigid Transformations

The goal of non-rigid surface registration methods proposed in the previous sections is to compute a transformation required to deform a source shape to a target shape. The pairwise non-rigid registration requires the designation of a target shape, to which the source shape is transformed. However, there are cases where correspondence information between multiple shapes is required. This framework is generally called groupwise non-rigid registration. In this framework, correspondence between multiple shapes is obtained by deforming all of the objects to a certain position. Generally, this position is the "average" position of all the shapes, as shown in Fig. 19. There have been many attempts to obtain this "average" shape, mainly through use of a template [9], [20], [28], optimization of transformations [2], [21], [30], and projection into another space [6], [18].

We claim that a meaningful groupwise registration should not only bring all the shapes in a group to an average location but also maintain the characteristic structures of each shape and align all of them to their average locations as well. This should be conducted by maintaining the overall accuracy as well as the feature matching accuracy.

The shapes in the group to be registered would be implicitly represented using SDF. From the SDFs of all the shapes, we estimate an "average" field. Although SDFs are not closed under operations such as addition and multiplication, in other words these operations on SDFs do not produce SDFs as a result, we avoid the problem by intentionally excluding some regions of the field that would cause instability during the averaging process and produce a "flat" average field.

The non-rigid registration of each shape is conducted using the

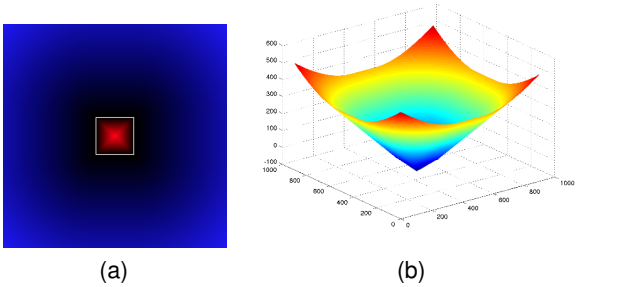



**Fig. 20** Visualization of signed distance field of a square. In 3D representation, the SDF has a conical shape, with a peak at the centermost part inside the square.

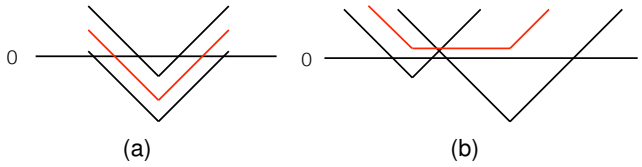



**Fig. 21** Relationship between the SDFs and the average. Black curves: shape SDFs. Red lines: direct average of SDFs. The location where the lines cross the horizontal line indicates the zero level set. If one surface completely surrounds the other surface, the average value is relatively stable. If they do not completely overlap, the average may produce a false or no zero level set, which is suppose to represent the surface of the average shape.

pairwise locally rigid globally non-rigid registration method proposed in the previous section. In this groupwise registration, each of the shapes in the group would be handled as a source shape that transforms to the target, the estimated average field.

We overcome the issue of ambiguity at the flat regions of the estimated field by utilizing the information in the local sampling regions of each of the source shapes. The exterior regions of the distance fields provide more information of the group of shapes compared to the regions within the shapes. The locally rigid globally non-rigid registration method takes advantage of the proposed average field by estimating the transformation of the sampling regions using the exterior distance information. By gradually deforming each of the shape to the estimated average field and refining the average field during the process, we achieve the registration of multiple shapes to the average location.

### 4.1 Average of Signed Distance Field

We propose to register a group of shapes by estimating the average field based on the SDFs of the shapes. However, as mentioned in the previous section, SDFs are not closed under operations such as addition and multiplication. This makes the calculation of and "average" SDF difficult. We first consider the issue using two shapes.

Fig. 20 shows SDF of a square. The SDF is also visualized in 3D. We set the initial shape so that the image region is 3 times as large as the input image, as was conducted in the previous section. As can be seen from the visualization in 3D, the SDF has a conical shape with a peak at the centermost part inside the shape.

Fig. 21 shows the 1D cross section of two pairs of SDFs and their average values. The black curves are the SDFs, and the red line indicates the average value of the SDFs. The location where the SDF curves crosses the horizontal line is the surface of the

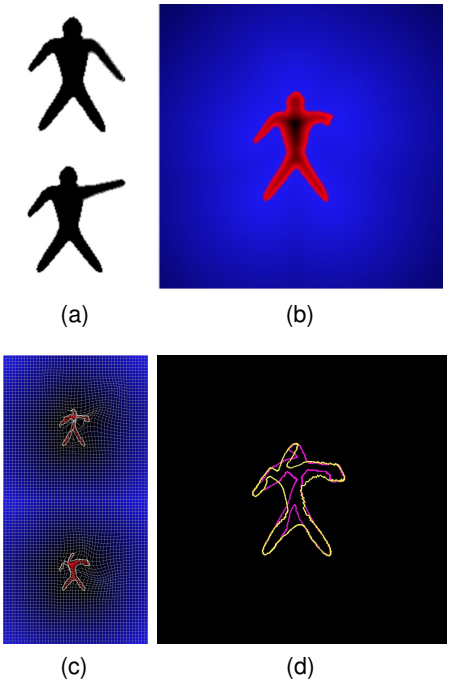



**Fig. 22** Results of registration using the average of SDFs. (a) Original images **A** and **B**. (b) Average value of SDFs. (c) Deformation on each object. (d) Registration result. The shapes fail to match at an optimal location.

shapes. If the shapes completely overlap each other with the peak at the same location, the average SDF can be used as the target for simultaneous groupwise registration. However, if the shapes do not overlap, which is the case in most registration scenarios, the average value may have some false or no zero level set, and the source shapes would be falsely registered to this value, as shown in Fig. 22. We can observe that the peaks of the SDFs are causing the estimation of the average to be significantly difficult.

### 4.2 Estimation of Registration Target

We propose to avoid the issue caused by the simple average values of the SDFs by removing the unstable regions from consideration. We achieve this by adding the absolute values of the SDFs to the average. In the case of two images, the proposed average becomes $(\mathbf{\Phi_A} + \mathbf{\Phi_B} + |\mathbf{\Phi_A}| + |\mathbf{\Phi_B}|)/4$. In the case of multiple shapes, this average can be defined as

$$\overline{\mathbf{\Phi}} = \frac{1}{2l} \sum_{i}^{l} (\mathbf{\Phi}_i + |\mathbf{\Phi}_i|), \tag{17}$$

where $l$ is the total number of shapes in the group.

Fig. 23 is the cross section of the proposed field. The modified average SDF is shown in In this average field, regions where all the shapes have positive SDF values would remain positive. In regions where all the shapes have negative SDF values, the value in the average field would be equal to 0.

In regions where SDF values are mixed, the absolute value of the SDF nullifies the effect of the negative values. This means that the effect of the shape with the negative value would be removed from the average. This is shown as regions surrounded by blue dotted lines in Fig. 23. This region is dependent on data with positive SDF values and therefore has more ambiguity compared to the exterior regions. However, as the shapes become

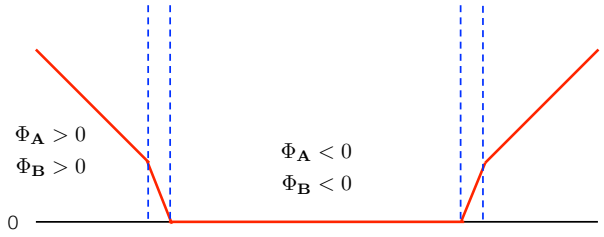



**Fig. 23** The proposed field calculated from two shapes **A** and **B** in 1D. The red line represents the average value. The interior regions of the shapes are flattened by adding the absolute values of the SDF.

closer together, this region is minimized and will disappear once the shapes are completely aligned. The locally rigid globally non-rigid strategy also eases the negative effect from this region. Most of the sampling regions would contain more exterior regions than these mixed regions. The rigid transformations calculated from the field within the sampling regions would be mainly based on the exterior information.

### 4.3 Locally Rigid Globally Non-rigid Registration with the Estimated Average

Using the method proposed in the previous section, we deform all the objects in a group to the proposed field obtained from all the SDFs. Here, the target would be $\overline{\mathbf{\Phi}}$ calculated from averaging the SDFs and the absolute distance fields. The error term and the smoothness term are expressed as

$$E_e = \int\int \alpha(x,y)\Big(\overline{\phi}\big(\mathbf{p}+\mathbf{t}(\mathbf{p})\big) - \phi_i(\mathbf{p})\Big)^2 d\mathbf{p} \qquad (18)$$

$$E_s = \lambda \int\int \Big(\mathbf{t}_x(\mathbf{p})^2 + \mathbf{t}_y(\mathbf{p})^2\Big) d\mathbf{p}\,, \qquad (19)$$

where $\mathbf{t} = (u, v)$ is the transformation of a point. The weight $\alpha(\mathbf{p}) = \frac{1}{1+e^{m(m-k)}}$ is the sigmoid that determines the importance relative to the shape surface, and $m = \frac{1}{2}(|\phi_i(\mathbf{p})| + |\overline{\phi}(\mathbf{p})|)$.

After discretization using the sampling points, the two Euler-Lagrange equation would be

$$\tfrac{1}{n}\alpha(\mathbf{p})\textstyle\sum_j^n \Big(\overline{\phi}(\mathbf{p}+\mathbf{s}_j+\mathbf{t}(\mathbf{p})) - \phi_i(\mathbf{p})\Big)\frac{\partial}{\partial \mathbf{t}}\overline{\phi}(\mathbf{p}+\mathbf{s}_j+\mathbf{t}(\mathbf{p}))$$
$$+\lambda\Big(\alpha_x(\mathbf{p})\mathbf{t}_x(\mathbf{p}) + \alpha_y(\mathbf{p})\mathbf{t}_y(\mathbf{p}(1-\alpha(\mathbf{p}))\Delta\mathbf{t}(\mathbf{p})\Big) = 0\,, \qquad (20)$$

where $\mathbf{p} = (x, y)$ represents the FFD control point and $\mathbf{s}_j = (s_j^x, s_j^y)$ is the sampling point in the sampling region surrounding the control point. From these equations, the translation of control points can be obtained as shown in the previous section.

Registration is also conducted in a hierarchical manner by subdividing the grid into half when registration reaches convergence in one layer. However, the convergence is determined by the difference between the proposed field and each SDF. When the difference between the error from the previous iteration to the current iteration becomes small, the method assumes that the calculation has converged in the layer. The same registration process is conducted for all $l$ shapes to register all the shapes in the group to the estimated average.

### 4.4 Extension to 3D data

The method can easily be extended to 3D data. This is straight forward as the proposed field can be calculated by obtaining the SDF of each surface and then averaging the SDFs and the absolute distance fields.

### 4.5 Experiment

We first evaluate the effectiveness of the proposed average field by conducting experiments using pairs of 2D contour images. We use the data from the image database provided by Sharvit *et al*. [24] and MPEG-7 CE-Shape-1 database [12]. We calculated the proposed field from the images, used it as a target and aligned each of the images by the method proposed in the previous section. The same setup as the previous experiments was used for the deformation, and maximum of 30 iterations were allowed for each layer in the hierarchical registration process.

#### 4.5.1 Overall Accuracy and Feature Correspondence

We compared the average shape obtained from our method to the average SDF calculated using the method by Chen *et al*. [6]. For fairness of comparison we used the pairwise non-rigid registration method proposed in the previous section to deform each shape to the average SDF obtained from the method by Chen *et al*. Initial settings, such as the initial image size, were set to be the same in both methods.

The experiments were conducted using “device”, “device2”, “fish”, “stef” and “dude”. The images of “fish”, “stef” and “dude” are from the database of Sharvit *et al*. [24], “device” and “device2” are from MPEG-7 CE-Shape-1 database [12].

Figs. 24- 27 show some of the registration results. Two images on the left are the input images with their given features. Images in third column are the registration results using the average SDF obtained from Chen *et al*. The images on the fourth column are the results from our proposed method. The red and the blue lines in the results indicate the original locations of the input shapes. The red surface after deformation is shown as a yellow surface, and the blue surface after deformation as pink surface. The feature points from the second image are shown as white points in order to see how close the feature points are aligned. The images on the right are the results from our method without the original shape information. Fig. 28 shows the average distance between the closest points and the corresponding feature points.

With “device 2” and “fish”, surface and feature matching had been achieved relatively accurately. However, the results using the average SDF of Chen *et al*. tend to be pulled toward one of the input images. On the other hand, our method estimated the final location accurately, and the final shapes are in between the two original surfaces without being pulled to either one of them.

The original images of “stef” and “dude” include structures that are far apart from each other. Registration results using the average SDF of Chen *et al*. succeeded in matching the similar regions, but could not determine the average position of structures with larger variance. We can observe that the average SDF obtained from the method by Chen *et al*. is accurate when shapes are very close. However, the results deteriorate when the corresponding structures do not overlap and are far from each other. Our method effectively matched the surfaces and features at a location in between the original shapes despite the large difference between the initial corresponding structures.

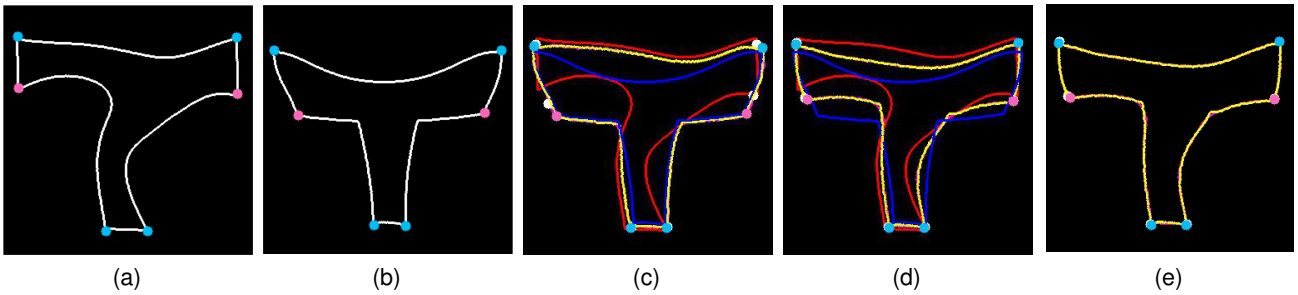



**Fig. 24** Experimental results of 2D contour data "device 2".

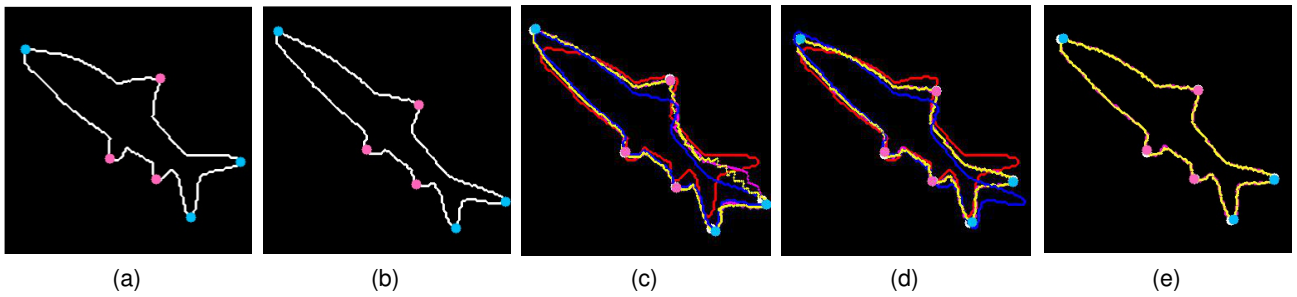



**Fig. 25** Experimental results of 2D contour data "fish".

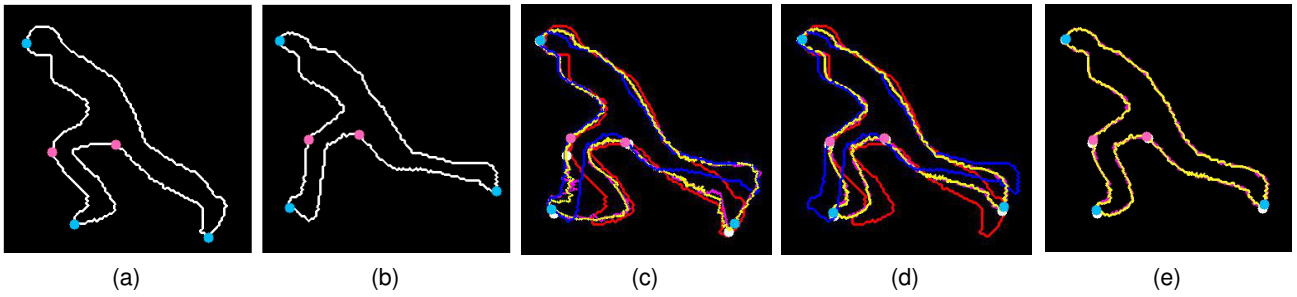



**Fig. 26** Experimental results of 2D contour data "stef".

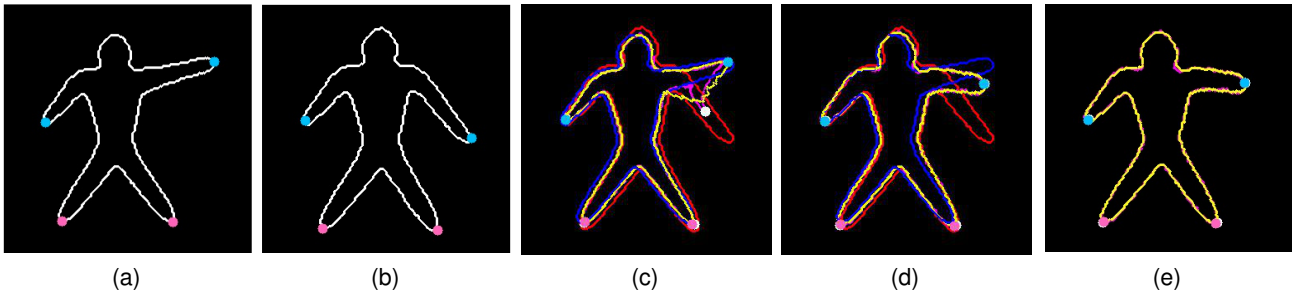



**Fig. 27** Experimental results of 2D contour data "dude". (a)(b) Input images. (c) Results using Chen *et al*. (d) Proposed method. (e) Proposed method without original surface. Red and blue surfaces in (c) and (d) represent the original shapes.

#### 4.5.2 Grid Movement

We then evaluated the method further by measuring the amount of FFD grid movement required to deform each image in the previous experiment. Figs. 29(a) and 29(b) show an example of FFD grid movement in the case of "fish". The average control point movement was recorded for each image.

Fig 29(c) shows the average grid movement in each of the cases. The two blue bars on the left side of each entry represent the grid movement of shape 1 and 2 using the average SDF from Chen *et al*. The two red bars on the right represents the grid movement of shape 1 and 2 using the proposed method.

We can observe that our method requires less amount of grid movement for each image. This shows that minimum amount of deformation was applied to each image in order to preserve the characteristic structures of the initial shapes. The grid movements in our method is also relatively balanced compared to the results using the average SDF of Chen *et al*., which indicates that the shapes have been registered to a position that requires a similar amount of deformation for each of the shapes.

#### 4.5.3 Simultaneous Registration of Multiple Images

To demonstrate that the method can handle more than 2 images, we then conducted experiments using 4 images of 2D contour

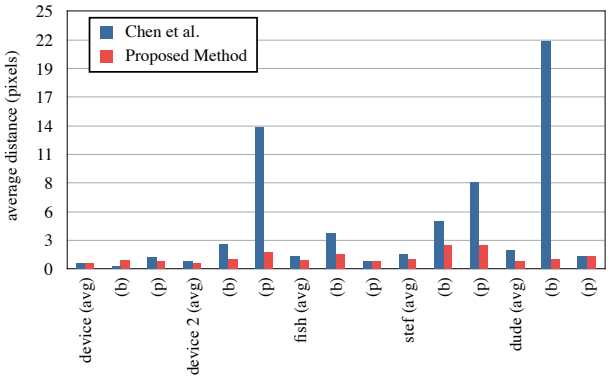



**Fig. 28** Overall accuracy and average distance between the corresponding features.

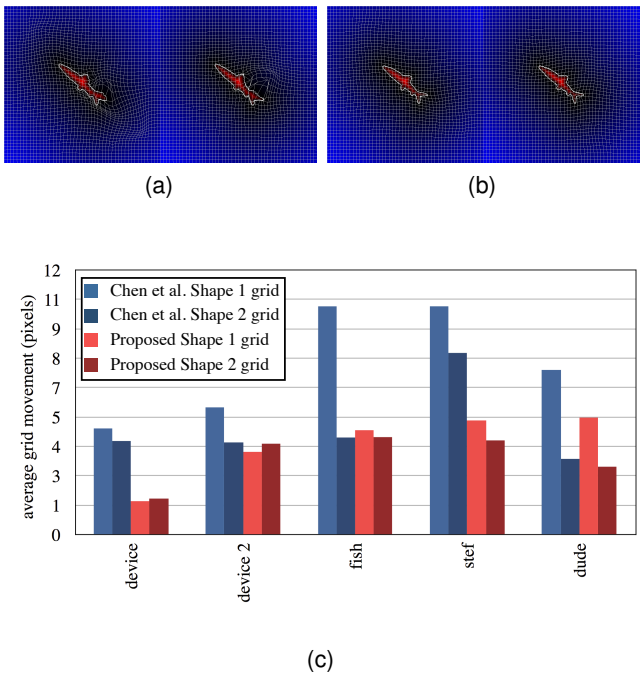



**Fig. 29** Comparison of FFD grid movement (a) Grid movement for "fish"using the average SDF of Chen *et al*. (b) Grid movement from the proposed method. (c) Comparison of average grid movement. Similar amount of deformation has been applied to each of the images in the pair.

data. We used "butterfly" and "dude" images from the database of Sharvit *et al*. [24]. In the same manner as the experiments using image pairs, we also obtained the average shape of the 4 images using our groupwise non-rigid registration framework. Maximum of 30 iterations were allowed for each layer in the hierarchical registration process.

Figs. 30 and 31 show the registration results of 4 "butterfly" and "dude" images. The top rows show all the input images with the blue and pink features. The bottom rows show the initial state before deformation and the registration result. Although the initial shapes and poses are quite different in the two cases, the method accurately gathered all the shapes to an average position without destroying the structure of each shape. All the prominent features were also aligned to the average position. The proposed groupwise non-rigid registration framework was able to align all

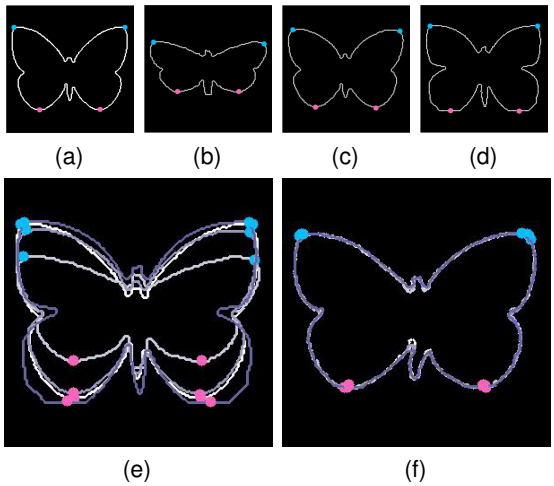



**Fig. 30** Registration results of "butterfly". (a) - (d) Initial images. (e) Before registration. (f) After registration.

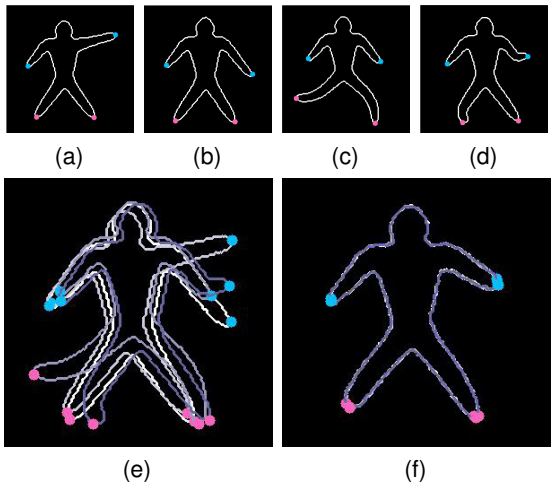



**Fig. 31** Registration results of "dude". (a) - (d) Initial images. (e) Before registration. (f) After registration.

the shapes to a meaningful position despite the larger number of input data. This has been done without manual intervention.

#### 4.5.4 Registration of 3D Data

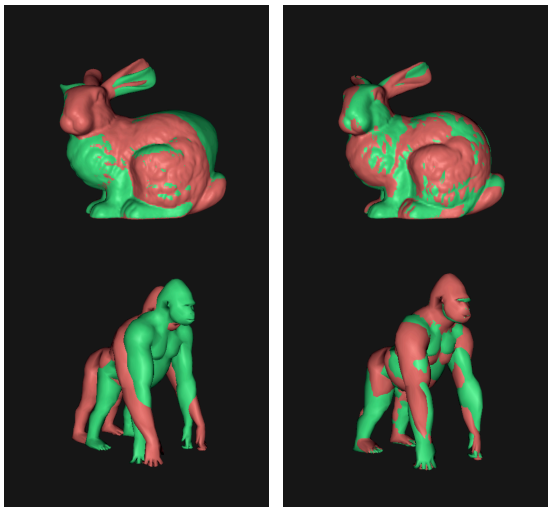
The method can easily be extended to 3D data. The proposed method just requires the third dimension $z$ to be added to the formulation. Fig. 32 shows the results using the Stanford Bunny and "gorilla" from the database of Bronstein *et al*. [3]. The shapes are accurately transformed to an average shape.

## 5. Conclusion

In this paper, we have proposed novel methods of non-rigid registration for shape analysis. In order to effectively align similar objects, we have proposed two methods based on the idea of "local rigidity and global deformability".

We first proposed a method based on a novel framework called dual-grid FFD. Similar to template matching, sampling regions try to find the best locations on the target that minimize the difference between the the partial source SDF within each sampling region and the entire target SDF and guide the overall deformation.

We then proposed another method that assigns a weight based on the distance from the surface and imposes a smoothness con-



(a) (b)

**Fig. 32** 3D data of Stanford Bunny and "gorilla" (a) before and (b) after registration

straint on the local rigid transformations, restricting radical movements that disrupt the deformation of the neighboring regions. This strategy provides flexibility to the proposed strategy, which enables a larger amount of deformation when there is a larger difference between the source and the target shapes.

To apply the same strategy for multiple objects simultaneously, we proposed a groupwise non-rigid registration method based on the locally rigid but globally non-rigid registration method. By estimating an average field from all of the SDFs of the shapes in a group, all of the shapes are registered to a single location by maintaining the characteristic structures. The average is estimated by adding the absolute distance value of the SDFs to intentionally avoid the peak information.

The experimental results of these methods applied to 2D contour data demonstrate the effectiveness of the "locally rigid globally non-rigid registration" strategy. All of these methods can easily be extended to 3D surface data just by adding the third dimension to all of the formulations. The results of the 3D data are also accurate, proving the effectiveness of these methods for analysis of surface data from various objects.

**Acknowledgments** This work was supported in part by National Science Foundation awards CAREER IIS-0746717 and IIS-0803670, and the Office of Naval Research grant N00014-11-1-0099 to KN, and Ministry of Education, Culture, Sports, Science and Technology under Digital Museum Project, and Strategic Information and Communications R&D Promotion Programme to KI.

## References

[1] P. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] K. Bhatia, J. Hajnal, B. Puri, A. Edwards, and D. Rueckert. Consistent groupwise non-rigid registration for atlas construction. In *IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pages 908–911, 2004.

[3] A. Bronstein, M. Bronstein, and R. Kimmel. Calculus of non-rigid surfaces for geometry and texture manipulation. *IEEE Transactions on Visualization and Computer Graphics*, 13:902–913, 2007.

[4] A. Bronstein, M. Bronstein, A. Bruckstein, and R. Kimmel. Analysis of two-dimensional non-rigid shapes. *International Journal of Computer Vision*, 78(1):67 – 88, 2008.

[5] L. Brown. A survey of image registration techniques. *ACM Computing Surveys (CSUR)*, 24(4):325–376, 1992.

[6] T. Chen, A. Rangarajan, S. Eisenschenk, and B. Vemuri. Construction of a neuroanatomical shape complex atlas from 3d mri brain structures. *Neuroimage*, 2012.

[7] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114 – 141, 2003.

[8] T. Gaens, F. Maes, D. Vandermeulen, and P. Suetens. Non-rigid multimodal image registration using mutual information. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1099–1106, 1998.

[9] X. Geng, G.E. Christensen, H. Gu, T.J. Ross, and Y. Yang. Implicit reference-based group-wise image registration and its application to structural and functional mri. *NeuroImage*, 47(4):1341, 2009.

[10] D. Hill, P. Batchelor, M. Holden, and D. Hawkes. Medical image registration. *Physics in Medicine and Biology*, 46(3):R1, 2001.

[11] X. Huang, N. Paragios, and D. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1303–1318, 2006.

[12] L. Latecki, R. Lakämper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 424–429, 2000.

[13] S. Lee, G. Wolberg, and S. Shin. Scattered data interpolation with multilevel b-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, 1997.

[14] B. Likar and F. Pernuš. A hierarchical approach to elastic registration based on mutual information. *Image and Vision Computing*, 19(1):33–44, 2001.

[15] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[16] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1 – 36, 1998.

[17] J. Pluim, J. Maintz, and M. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986 – 1004, 2003.

[18] K.M. Pohl, J. Fisher, S. Bouix, M. Shenton, R.W. McCarley, W.E.L. Grimson, R. Kikinis, and W.M. Wells. Using the logarithm of odds to define a vector space on probabilistic atlases. *Medical Image Analysis*, 11(5):465, 2007.

[19] K. Pulli. Multiview registration for large data sets. In *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*, pages 160–168, 1999.

[20] D. Rueckert, A. Frangi, and J. Schnabel. Automatic construction of 3d statistical deformation models using non-rigid registration. In *Proceedings of International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 77–84, 2001.

[21] M. Rumpf and B. Wirth. An elasticity approach to principal modes of shape variation. *Scale Space and Variational Methods in Computer Vision*, pages 709–720, 2009.

[22] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152, 2001.

[23] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 151–160, 1986.

[24] D. Sharvit, J. Chan, H. Tek, and B. Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 9(4):366–380, 1998.

[25] R. Sumner and J. Popovic. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23(3):399–405, 2004.

[26] R. Szeliski and S. Lavallée. Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. *International Journal of Computer Vision*, 18(2):171–186, 1996.

[27] G. Turk and M. Levoy. Zippered polygon meshes from range images. *Proceedings of ACM SIGGRAPH*, pages 311 – 318, 1994.

[28] G. Wu, H. Jia, Q. Wang, and D. Shen. Sharpmean: Groupwise registration guided by sharp mean image and tree-based registration. *NeuroImage*, 56(4):1968, 2011.

[29] C. Yang and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[30] A.J. Yezzi and S. Soatto. Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. *International Journal of Computer Vision*, 53(2):153–167, 2003.

[31] B. Zitová and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977 – 1000, 2003.