

Published in final edited form as:

*IEEE Trans Image Process.* 2012 October ; 21(10): . doi:10.1109/TIP.2012.2206034.

## Filtering in the Diffeomorphism Group and the Registration of Point Sets

**Yi Gao,**

Department of Psychiatry, Brigham and Women's Hospital, Harvard Medical School, Boston, MA 02115 USA

**Yogesh Rathi, IEEE [Member],**

Department of Psychiatry, Brigham and Women's Hospital, Harvard Medical School, Boston, MA 02115 USA

**Sylvain Bouix, IEEE [Member], and**

Department of Psychiatry, Brigham and Women's Hospital, Harvard Medical School, Boston, MA 02115 USA

**Allen Tannenbaum, IEEE [Fellow]**

Departments of Electrical and Computer Engineering and Biomedical Engineering, Boston University, Boston, MA 02115 USA

Yi Gao: gaoyi@bwh.harvard.edu; Yogesh Rathi: yogesh@bwh.harvard.edu; Sylvain Bouix: sylvain@bwh.harvard.edu; Allen Tannenbaum: tannenba@bu.edu

### Abstract

The registration of a pair of point sets as well as the estimation of their pointwise correspondences is a challenging and important task in computer vision. In this paper, we present a method to estimate the diffeomorphic deformation, together with the pointwise correspondences, between a pair of point sets. Many of the registration problems are iteratively solved by estimating the correspondence, locally optimizing certain cost functionals over the rigid or similarity or affine transformation group, then estimating the correspondence again, and so on. This type of approach, however, is well-known to be susceptible to suboptimal local solutions. In this paper, we first adopt the perspective of treating the registration as a posterior estimation optimization problem and solve it accordingly via a particle-filtering framework. Second, within such a framework, the diffeomorphic registration is performed to correct the nonlinear deformation of the points. In doing so, we provide a solution less susceptible to local minima. We provide the experimental results, which include challenging medical data sets where the two point sets differ by 180° rotation as well as local deformations, to highlight the algorithm's capability of robustly finding the more globally optimal solution for the registration task.

### Index Terms

Particle filters; point set deformable registration; polyaffine transformation

### I. Introduction

Matching/Registering two sets of points is a problem that finds its applications in various area in computer vision [1]–[5]. In general, the registration problem consists of two

intertwined parts. First, the correspondence between the points in the two sets must be identified explicitly or implicitly. Second, a proper transformation, linear or nonlinear, is computed to minimize a certain metric that measures the discrepancy between the two point sets under the current correspondence relationship.

In cases where the correspondence is already known, and the transformation is assumed to be similarity (rigid motion plus isotropic scaling), a closed-form solution was given by Horn in [6]. However in more general cases where either the correspondence is unknown or the transformation takes more complicated forms, an analytical result rarely exists. The authors of [1], [2] introduced the well-known Iterative Closest Point (ICP) method to approach the problem by an iterative two step scheme. First the point-wise correspondence is identified, and then the rigid transformation parameters are estimated by minimizing the summed squared Euclidean distances between the corresponding point pairs. Other researchers infer the correspondence among points from the homologous spatial structures or features [7]. The original ICP algorithm is known to be sensitive to initialization and susceptible to local minima. To address such issues, Fitzgibbon in [8] modifies the optimization procedure by using the Levenberg-Marquardt method to speed up as well as to obtain a wider region of convergence. On the other hand, in this papers [9], [10] the corresponding point is selected by utilizing the invariant features together with the spatial position. Moreover, the initialization difficulty is addressed in [11], [12] where the authors use a deterministic annealing for optimization and a neural network model for correspondence. More recently, Li and Hartley solve the problem under a Lipschitz optimization framework, yielding a global optimal rigid transformation in the special Euclidean group [13].

Moreover, due to the noise of the two (or more) point sets, some researchers have chosen to employ inexact correspondences between the point sets. For instance, in the different but highly related landmark image registration field, the landmarks extracted (manually or automatically) from the two images may not be perfectly match one another. To solve that problem, Bookstein has applied a linear regression model in the interpolation of a spline-based transformation vector field [14], whereas Christensen *et al.* incorporate image intensity information into the landmark matching under a hierarchical fluid model [15]. More recently, Rohr *et al.* presented an algorithm to regularize both affine and thin-plate-spline components of the transformation in a functional minimization framework [16]. Along a similar lines, for the case of the registration of point sets, while points can be treated as a cloud in  $\mathbb{R}^3$ , considering them as samples from a larger object may exploit the more global information of the object's shape. Accordingly, the authors of [17]–[21] first extract from the point set certain global geometric objects such as curves or surface patches (called “shape descriptors”), and then perform registration on them. It is shown that the performance over poor initialization and the partial structure is improved by a such scheme.

Other researchers define the correspondences implicitly. For example, in [22], Tsin and Kanade adopted the idea of image correlation in order to define an affinity measurement for point sets by estimating the probability distribution functions from point sets and computing the “kernel correlation” between the two point sets by using kernel functions. Consequently, the point sets are registered by maximizing the kernel correlation between the two distributions. In [23], Jian and Vemuri model the point sets by Gaussian mixtures and derive a closed form solution for minimizing the  $L^2$  distance between the Gaussian mixtures. Closely related to this approach, Wang *et al.* model the point sets by Gaussian mixtures, and the Jensen-Shannon divergence is used as the (dis-)similarity measurement to perform non-rigid registration [24]. Granger and Pennec propose a multi-scale generalized maximum-likelihood algorithm, called EM-ICP, to estimate the rigid transformation as well as the point correspondence under the Expectation-Maximization framework [25]. Furthermore, Myronenko and Song maximize the likelihood between the Gaussian mixture models

representation of the two point sets coherently with the topological structure of the point sets [26]. Apart from mixture of Gaussians, Glaunes *et al.* represent points by delta functions, together with the surface information, the point sets are aligned using the dual norm in a reproducing kernel Hilbert space [27].

### A. Related Work and Motivation

One common feature of the aforementioned methods lies in the fact that they treat the optimal transformation parameters as fixed (but unknown) values, and various (greedy/local) optimization procedures are used to find them. However, from a different perspective, the parameters may also be treated as a (multi-dimensional) random variable or as a state variable of the accordingly designed dynamical system. Formulating the problem in this manner enables one to exploit the more global aspects of registration via a parameter estimation/filtering based scheme. To this end, the authors of [28]–[31] use the particle filter or unscented particle filter for a rigid surface registration. Along these lines, Moghari and Abolmaesumi adopted the unscented Kalman filter to perform a rigid registration for point sets [32].

However, although the filtering schemes yield more globally satisfying registration results, the transformations are limited to rigid or similarity transformations which only have 6 or 7 degree of freedom. Indeed, most of these registration-by-filtering methods contain the idea of stochastically exploring a large part of the state space. Hence, in order for the problem to be practically feasible, the dimensionality of the state space cannot be too high. This may be a key obstacle for these methods to be applied in deformable registration, where the transformation belongs to an inherently infinite (or very high) dimensional transformation space. In fact, local optimal nonlinear/deformable registration is only addressed in a few papers, such as [24], [26], [27], [33]. Note that due to the difficulty of high dimensionality, directly extending the filtering scheme to the deformable registration is computationally intractable. Hence, one needs to consider new approaches in order to exploit the more global aspect of the filtering schemes in the deformable registration scenarios, and this is the main objective of the present work.

Moreover, in the different but highly related image registration field, Pitiot proposed a piece-wise affine registration scheme [34]. Since the large transformation/distortion is not a severe problem for images in this latter work, registration is only carried out by a local optimization process. Furthermore, the final transformation obtained from the affine sub-transformations is by erosion and interpolation around the sub-image boundaries. In contrast, the final result of the method proposed in this paper is guaranteed to be diffeomorphic. One should also note that although a deformable transformation may provide greater flexibility in matching the point sets, it may also cause degeneration problems where the transformation may tear apart or glue together certain parts of the point sets. Hence, we require the deformable transformation to be diffeomorphic. Nevertheless, there are cases in which a non-diffeomorphic transformation is necessary, for instance when registering a disk to a torus. However in this work, we assume there exists a meaningful diffeomorphism between the two large enough point sets (so that the problem is not under-determined), and the objective of the present work is to locate it in the solution space.

### B. Contribution of This Work

In this paper, we propose a natural method for filtering in an infinite dimensional diffeomorphism space, and (diffeomorphic) deformable registration in the filtering framework. The registration result inherits several nice properties from state estimation/filtering theory. Further, we adopt a multi-scale scheme so that the filtering in the deformable transformation state space is locally attained by an affine transformation and is

thus computationally feasible. The final transformation is a diffeomorphism, which is not always guaranteed by most of the spline based transformations.

In general, we first approach the registration through a filtering scheme in the affine transformation group. Then the point set is hierarchically decomposed, and the registration is performed in a finer scale to address the local deformations. In each registration-by-filtering step, the dimension of the state space is always 12 (affine). Further, the transformations at local levels are then fused together in order to form a smooth and smoothly invertible (diffeomorphic) transformation. In short, by carrying out this process, we cast the deformable registration task into a filtering framework with its globally more satisfiable performance, feasible computation load, with resulting transformation being a diffeomorphism.

The remainder of this paper is organized as follows. In Section II, we present the general particle filtering scheme and show its capability of avoiding local minima. Then, in Section III, we present the main algorithm of multi-scale polyaffine registration under the particle filtering framework, with the experiments conducted in Section IV. Finally, the ongoing and future work is discussed in Section V.

## II. Optimization via Particle Filtering

Avoidance of local minima in optimization is an important yet sometimes difficult problem. If the problem to be solved has certain property such as being quadratic, convex, etc., then it can be solved with efficient algorithms [35]. Unfortunately, without any *a priori* knowledge of the optimization problem, no algorithm is better than the blind search [36]. In this section we first briefly review the general framework for the particle filtering. Then, a general method for optimization is proposed under the particle filtering framework.

### A. Particle Filtering (PF)

We very briefly summarize particle filtering here. We do not give the most general formulation of the method, but simply of sufficient generality for the purposes of this paper [37].

Given a dynamic system, the objective of particle filtering is to estimate the state variables of the system. Formally, the state variable at time  $t$  is denoted as a vector  $\mathbf{a}_t \in \mathbb{R}^n$ . The transition function governing the evolution of the state variable,  $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ , is defined as

$$\mathbf{a}_{t+1} = f(\mathbf{a}_t, \mathbf{u}_t) \quad (1)$$

in which the process noise  $\mathbf{u}_t \in \mathbb{R}^m$  is white and independent of the state variable. Moreover, the observation at time  $t$ ,  $\mathbf{b}_t \in \mathbb{R}^r$ , is obtained by the observation model

$$\mathbf{b}_t = g(\mathbf{a}_t, \mathbf{v}_t) \quad (2)$$

where the observation noise  $\mathbf{v}_t$  is white and independent of both the system state and the process noise. Since the underlying state variables are not directly observable, particle filtering is a sequential Monte Carlo method [37] that provides a sequential estimate of the posterior distribution of  $\mathbf{a}_t$ , based on all the observations made until time  $t$ ,  $\mathbf{b}_{1,2}, \dots, \mathbf{b}_t$  ( $\mathbf{b}_{1:t}$ ), namely  $p(\mathbf{a}_t | \mathbf{b}_{1:t})$ . In contrast to the requirements of the celebrated Kalman filter, here the  $f$  and  $g$  do not have to be linear functions, nor do the  $\mathbf{u}_t$  and  $\mathbf{v}_t$  have to be additive and Gaussian. All the particle filtering requires are the functions  $f$  and  $g$ , probability density functions (PDF)  $p(\mathbf{u}_t)$ ,  $p(\mathbf{v}_t)$ , and  $p(\mathbf{a}_0)$ .



We now have to show how the algorithm estimates  $p(\mathbf{a}_{t+1}|\mathbf{b}_{1:(t+1)})$  given  $p(\mathbf{a}_t|\mathbf{b}_{1:t})$ . To achieve this goal in the Monte Carlo framework, assume we have the current samples (particles) for  $p(\mathbf{a}_t|\mathbf{b}_{1:t})$  as  $\{\mathbf{a}_t^{(i)}\}_{i=1}^R$ . In order to obtain the samples for  $p(\mathbf{a}_{t+1}|\mathbf{b}_{1:(t+1)})$ , we first generate the *prediction particles*,  $\{\tilde{\mathbf{a}}_{t+1}^{(i)}\}_{i=1}^R$  by passing the current samples through the system's process model. That is,  $R$  random samples  $\{\mathbf{u}_t^{(i)}\}_{i=1}^R$  are generated from process noise  $\mathbf{u}_t$  and we further compute

$$\tilde{\mathbf{a}}_{t+1}^{(i)} := f(\mathbf{a}_t^{(i)}, \mathbf{u}_t^{(i)}), i=1, \dots, R. \quad (3)$$

It is noted that the above generation of the *prediction particles* is based solely on the information up to time  $t$  and can thus be computed before time  $t+1$ . Furthermore, when the observation  $\mathbf{b}_{t+1}$  is available at time  $t+1$ , the *prediction particles* can then be updated to reflect the information brought in by the new observation. Formally, first the weight of each *prediction*

### Algorithm 1

#### Simulated Annealing Algorithm

- 
- 1: Start from  $\mathbf{a}_0$
  - 2: **for** Optimization criteria reached **do**
  - 3:    $\tilde{\mathbf{a}}_{t+1} = \text{Candidate}(\mathbf{a}_t)$
  - 4:   Accept  $\mathbf{a}_{t+1} = \tilde{\mathbf{a}}_{t+1}$  with probability  $\min\{1, \exp((C(\mathbf{a}_t) - C(\tilde{\mathbf{a}}_{t+1}))/\text{Temperature}(t))\}$
  - 5: **end for**
- 

*particles*,  $\xi^{(i)}$ , is computed as

$$\xi^{(i)} = \frac{p(\mathbf{b}_{t+1}|\tilde{\mathbf{a}}_{t+1}^{(i)})}{\sum_{j=1}^R p(\mathbf{b}_{t+1}|\tilde{\mathbf{a}}_{t+1}^{(j)})}, i=1, \dots, R \quad (4)$$

in which the likelihood term

$$p(\mathbf{b}_{t+1}|\tilde{\mathbf{a}}_{t+1}^{(i)}) = \int \delta(\mathbf{b}_{t+1} - g(\tilde{\mathbf{a}}_{t+1}^{(i)}, \mathbf{v}_{t+1})) \times p(\mathbf{v}_{t+1}) d\mathbf{v}_{t+1} \quad (5)$$

can be computed by Monte Carlo simulation on the observation noise  $\mathbf{v}_{t+1}$ .

Lastly, the *posterior particles*  $\{\mathbf{a}_{t+1}^{(i)}\}_{i=1}^R$ , which represent the posterior PDF  $p(\mathbf{a}_{t+1}|\mathbf{b}_{1:(t+1)})$ , are generated from the *prediction particles* by using a re-sampling scheme. More explicitly, for each  $i = 1, \dots, R$ , we generate a random sample  $\mu_i$  from the uniform distribution on  $(0, 1]$ . Then, the *prediction particles*  $\tilde{\mathbf{a}}_{t+1}^{(J)}$  are chosen as  $\mathbf{a}_{t+1}^{(i)}$  where  $J$  satisfies:

$$\sum_{j=0}^{J-1} \xi^{(j)} < \mu_i \leq \sum_{j=0}^J \xi^{(j)} \quad (6)$$

with  $\xi^{(0)} = 0$ . By doing so, we obtain the samples  $\{\mathbf{a}_{t+1}^{(i)}\}_{i=1}^R$  for the posterior pdf  $p(\mathbf{a}_{t+1}|\mathbf{b}_{1:(t+1)})$ . As a result, the algorithm starts with samples of  $p(\mathbf{a}_0)$  and recursively generates the samples for  $p(\mathbf{a}_t|\mathbf{b}_{1:t})$  for any  $t$ .

## B. Local Minima Avoidance via PF

The simulated annealing (SA) algorithm is a widely used global optimization procedure [38]. In order to optimize (minimize from now on) a cost function  $C : \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}$ , on the state space  $\Omega$ , the basic steps of the SA algorithm is described as shown in Algorithm 1.

As time  $t \rightarrow \infty$ , the SA algorithm converges to the global minimal of  $C$  [38]. Next, we show how SA can be cast into the PF framework. In fact, if we define

$$\begin{cases} \mathbf{a}_{t+1} = \mathbf{a}_t + \mathbf{u} \\ \mathbf{b}_t = C(\mathbf{a}_t) + v \end{cases} \quad (7) \quad (8)$$

where  $v$  is the scalar observation noise with negative exponentially distribution. Then, applying particle filtering on this system is equivalent to SA with  $\text{Candidate}(\mathbf{a}_t) = \mathbf{a}_t + \mathbf{u}$ . Moreover, since there is no physical observation in this optimization-by-filtering scheme, its definition is left as another degree of freedom. One possibility is to define it as  $\min\{\tilde{\mathbf{b}}_t^{(i)}\}_{i=1}^R$ , with  $\tilde{\mathbf{b}}_t^{(i)} := C(\mathbf{a}_t^{(i)})$  where  $\{\mathbf{a}_t^{(i)}\}_{i=1}^R$  are the particles. Indeed, the dynamics and filtering schemes employed in [30]–[32], [39] fall into this general formulation.

However, the above identity process model does not contribute to the overall optimization process because it performs no optimization evolution. Indeed, the optimization capability of filtering such systems is mainly due to the selection of the optimal state during observation. Therefore, it is a natural requirement to re-design the process model such that the system will autonomously evolve to states with lower costs. As a result, the particles will be more concentrated around the minima than in the identity process model case. To this end, we re-define the dynamic system as:

$$\begin{cases} \mathbf{a}_{t+1} = \mathcal{O}(\mathbf{a}_t + \mathbf{u}_t) \\ \mathbf{b}_t = C(\mathbf{a}_t) + v_t. \end{cases} \quad (9) \quad (10)$$

In equation (9),  $\mathcal{O} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is a deterministic optimization operator which brings the (perturbed) input state  $\mathbf{a}_t + \mathbf{u}_t$  to a new state  $\mathbf{a}_{t+1}$  with lower cost. For instance,  $\mathcal{O}(\mathbf{a}) := \mathbf{a} - \Delta t \cdot \nabla C(\mathbf{a})$ . In doing so, the dynamic system will evolve by itself to lower cost states, which accelerates the optimization process. Moreover, this is equivalent to SA with  $\text{Candidate}(\mathbf{a}_t) = \mathcal{O}(\mathbf{a}_t + \mathbf{u})$ , which is often employed in the SA context.

With the general optimization-by-filtering given here, in the next section, we will formulate point set diffeomorphic registration problem as that of filtering a dynamic system of the form given in equation (9) and (10), and solve it using particle filtering.

## III. Point Set Diffeomorphic Registration

There are certain issues involved by simply employing a given local optimal registration scheme in equations (7) or (9), and then estimating the system state using filtering technique in order to provide a more globally satisfying result.

Indeed, this is the main idea in [30]–[32], [39] where the transformation is limited to only rigid or similarity transformations. However, if such scheme is to be applied directly to deformable registration, due to its many more degrees of freedom, the computational burden may be very large especially in 3D case. Many previous deformable registration schemes parametrize the deformation by using splines or a point-wise vector field, in which case the

dimension of the appropriate variable space can easily be more than  $10^3$  (10 control points along each dimension).

Furthermore, not only may the the computation load be prohibitively high, but in some cases many of the degrees of freedom may not be necessary. For example, in registering robotic arms, we may face the situation where an “L” shaped joint is to be registered with a “V” shaped one. In such a case, instead of a full set of spline nodes controlling the deformation, two sets of rigid transformations, each transforming one arm, will be sufficient. Therefore, the degrees of freedom for deformable transformations should be adaptive. In order to achieve that goal, and apply the filtering technique in deformable registration, we note that affine transformations are parametrized by 12 parameters, and further any deformable registration transformation that is a diffeomorphic is locally affine, which is locally represented by its Jacobian matrix. Hence, we propose a method to perform diffeomorphic deformable registration by adaptively and hierarchically performing particle filtering affine registration at multiple scales.

In this section, we present the theoretical details of the diffeomorphic registration-by-filtering scheme, in order to register the “moving point set” to the “fixed point set”. More explicitly, the algorithm begins with a single affine registration using a particle filtering technique. In addition, the resulting moving point set is further decomposed if it does not meet the pre-defined accuracy criteria, and registration is performed at finer scale to correct the local deformation. Such decomposition-registration cycle continues until satisfying result is achieved. Finally, under a Lie-group/Lie-algebra framework, a diffeomorphic transformation is constructed as the final result.

### A. Affine Registration Using PF

As the initial step for deformable registration, we utilize the particle filtering method for affine group of transformations. The significance of this step is two fold: First, any diffeomorphic transformation is locally affine; second, in contrast to the dynamic system proposed in [30]–[32], [39], the process model in equation (9) is more efficient in achieving the optimality. Formally, denote the fixed point set as  $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_M\}$  and the moving point set, which is to be registered to the fixed point set, as  $\mathcal{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_N\}$ . It is noted that  $\mathbf{q}_i, \mathbf{p}_i \in \mathbb{R}^3 \times \{1\}$  are represented in homogeneous coordinates. To register  $\mathcal{Q}$  to  $\mathcal{P}$  a particle filtering affine registration procedure is carried out. For this purpose, we define the cost function

$$C(\mathbf{B}) := \frac{1}{N} \sum_{i=1}^N \|\mathbf{e}_i(\mathbf{B})\|^2 + \frac{\lambda}{\det^2(\mathbf{A})} \quad (11)$$

$$:= \frac{1}{N} \sum_{i=1}^N \|\mathbf{B}\mathbf{q}_i - \mathcal{Cl}(\mathbf{B}\mathbf{q}_i)\|^2 + \frac{\lambda}{\det^2(\mathbf{A})}. \quad (12)$$

In equation (12), we have the transformation matrix

$$\mathbf{B} = \begin{pmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}_{1 \times 3} & \mathbf{1} \end{pmatrix} \quad (13)$$

where  $\mathbf{A} \in \mathbb{R}^{3 \times 3}$ ,  $\det(\mathbf{A}) \neq 0$  is the affine transformation matrix, and  $\mathbf{t} \in \mathbb{R}^3$  is the translation vector. In addition,  $\mathcal{Cl} : \mathbb{R}^3 \rightarrow \mathcal{P}$  maps a point in  $\mathbb{R}^3$  to its closest point in  $\mathcal{P}$ . Note that the second term in equation (12) acts as a regularizer. With weighting  $\lambda > 0$ , it penalizes  $\det(\mathbf{A})$  from getting too close to zero. Without this, the cost function may have trivial minimizers. It is noted that such scale factor does affect the cost. However, in non-degenerate situations,

the magnitude of the first term in the cost function usually dominates. Only in the degenerate situations where  $\det(A)$  approaches zero, its reciprocal goes to infinity and this term then plays a non-negligible role in the cost function. So we usually adjust the  $\lambda$ , to for example  $10^{-3}$ , so that the second term is small relative to the first term.

Concerning numerical details, a KD-tree data structure is utilized to achieve a fast ( $O(\log M)$ ) search [40]. The minimization of the registration cost function  $C$  in (12) is a

### Algorithm 2

#### Affine Registration by Particle Filtering

---

```

1:
    Sample from  $p(a_0)$  to get particles  $\{a_0^{(i)}\}_{i=1}^R$ 
2: for  $i = 1, 2, \dots$  (converge or reach max iterations) do
3:
    Obtain the prediction particles  $\{\tilde{a}_t^{(i)}\}_{i=1}^R$  by applying the process model on the previous particles  $\{a_{t-1}^{(i)}\}_{i=1}^R$ 
4: Evaluate the likelihood  $\xi^{(i)}$ 's using (4)
5:
    Resample to get posterior particles  $\{a_t^{(i)}\}_{i=1}^R$  using equation (6)
6: end for

```

---

12 dimensional nonlinear unconstrained optimization problem. In order to minimize  $C$ , the BFGS algorithm (one type of quasi-Newton methods [35]) is employed to obtain a fast (super-linear) convergence. Moreover, in computing the gradient, the image of the mapping  $Cl$  in equation (12) is considered to be fixed in each ICP optimization step. So the differentiability issue of the  $Cl$  mapping is thereby handled.

Nevertheless, the above affine ICP is a local minimization process and is known to be sensitive to initialization and local minima. To overcome such drawbacks and achieve a more global optimality, we pose such optimization problem under the filtering scheme introduced in Section II-B, and solve it using particle filtering. Accordingly, we explicate next the definition of the state space and the process/observation models in equations (9) and (10).

In affine registration, the state space  $\mathcal{S}$  is 12 dimensional where the first 9 dimensions are for the affine matrix and the last 3 dimensions are for translation. In the given state space, denote the state vector as  $a_t \in \mathcal{S}$ . Then, the process model operator  $\mathcal{O}: \mathcal{S} \rightarrow \mathcal{S}$  in equation (9) is formulated as “taking  $a_t$  as the initial configuration, proceeding with a few steps of the deterministic affine registration (with respect to the cost function  $C$  in equation (12)), and returning the resulting configuration as  $a_{t+1}$ .” Moreover, in all the tests below, the number of steps for the deterministic optimization is fixed to 3, to balance the accuracy and computational load. Furthermore, the process and observation noise terms,  $u_t$  and  $v_t$  are assumed to be independent with time  $t$ . Finally, the observation model in equation (10) is adopted from equation (12). It can be seen that both the process and the observation models in equations (9) and (10) are highly non-linear. This again justifies the necessity of the particle filtering scheme.

In addition to the process and observation models, the prior distribution  $p(x_0)$  is required. Usually, without *a priori* information of the solution,  $p(x_0)$  is assumed to be Gaussian or uniformly distributed [37], [41]. Altogether, the complete particle filtering affine ICP can be

described in Algorithm 2. This algorithm will be referred to as affine-PF-ICP in the subsequent discussion.

It is noted that the proposed algorithm focuses mainly on a way to perform filtering in the high-dimensional (theoretically infinite dimensional) diffeomorphic transformation group and it is “compatible” with any outlier rejection method, such as RANSAC [42]. In the current form, the registration cost function is relatively robust to the outliers in the moving point set since it only traverses the fixed point set. Yet, if there are many outliers in the fixed point set, the current algorithm may perform sub-optimally. Nevertheless, any outlier rejection method can be used in conjunction with the proposed filtering scheme to improve the robustness to the outlier.

## B. Transformation Decomposition

The particle filtering affine registration presented in the previous section yields an affine transformation  $\tilde{\mathbf{B}} \in \mathbb{R}^{4 \times 4}$ . Although such an affine registration result is achieved by particle filtering that is much less sensitive to local minima, and is thus capable of handling the cases where two point sets differ by even an 180° rotation (around one or more axes), it is not able to provide satisfying results where the local deformations exist.

Indeed, mentioned in Section III, the deformations should be addressed in a multi-scale fashion. In order to decide if the current registration is satisfying, and therefore whether the registration should further be pursued in a finer scale, we measure the registration error at each point. That is, the error vector term  $\mathbf{e}_i$  as in equation (12) is used:

$$\mathbf{e}_i(\tilde{\mathbf{B}}) := \mathbf{p}_j - \tilde{\mathbf{B}}\mathbf{q}_i := Cl(\tilde{\mathbf{B}}\mathbf{q}_i) - \tilde{\mathbf{B}}\mathbf{q}_i \quad (14)$$

where  $\mathbf{q}_j$  is the closest point in  $\mathcal{P}$  to  $\tilde{\mathbf{B}}\mathbf{q}_i$ . Accordingly, we define the average error magnitude

$$E := \frac{1}{N} \sum_{i=1}^N \|\mathbf{e}_i(\tilde{\mathbf{B}})\|^2 \quad (15)$$

which is consistent with the cost function in equation (12) but without the regularization term.

Such a quantity is used to determine if the registration error is less than the pre-specified tolerance. If not, the moving point set is decomposed in a way described below. To proceed, each point in the registered moving point set is augmented using its error vector defined above to form a 8-dimensional feature vector

$$\mathbf{f}_i := [\eta\mathbf{q}_i; (1-\eta)\mathbf{e}_i] \in \mathbb{R}^8. \quad (16)$$

It is noted that since the 4-th component of  $\mathbf{q}_i$  and  $\mathbf{e}_i$  are respectively 1 and 0,  $\mathbf{f}_i$  is effectively in  $\mathbb{R}^6$ . By doing this, the feature vector combines the original space information of the point set as well as the registration error. In practice,  $\eta$  is used to balance the contributions of the two halves of the feature vector  $\mathbf{f}_i$ 's. Formally, it is computed as an integer power of 10 such that

$$\left\lceil \log_{10} \left( \eta \sum_{i=1}^N \|\mathbf{q}_i\| \right) \right\rceil = \left\lceil \log_{10} \left( (1-\eta) \sum_{i=1}^N \|\mathbf{e}_i\| \right) \right\rceil. \quad (17)$$

Furthermore, a sparse graph is constructed on the  $N$  feature vectors. The sparsity of the graph is controlled in such a manner that for each vertex, there are only few closest vertices

connected to it (in all the experiments, the 10 closest vertexes are chosen). Moreover, on the edge connecting  $f_i$  and  $f_j$ , the weight is  $W(i, j) := \exp(-\|f_i - f_j\|^2)$ . Therefore, the (dis-)similarity matrix  $W \in \mathbb{R}^{N \times N}$  is very sparse. It is noted that the construction of the matrix  $W$  does not take  $O(N^2)$  temporally or spatially. In fact, in the feature space a KD-tree is constructed. Hence the time complexity for constructing  $W$  matrix is  $O(N \log N)$ .

On this graph  $G(Verts, Edges)$ , we seek to find a partition that divides  $G$  into two sub-parts  $U_1, U_2 \subset Verts$ , with  $U_1 \cap U_2 = \emptyset$ ,  $U_1 \cup U_2 = Verts$ , and minimizing the normalize-cut energy  $Ncut(U_1, U_2)$  defined as

$$Ncut(U_1, U_2) := \sum_{j=1}^2 \left( \frac{\sum_{f_1 \in U_1, f_2 \in U_2} W(f_1, f_2)}{\sum_{f_j \in U_j, f \in Verts} W(f_j, f)} \right). \quad (18)$$

In order to minimize this energy, the normalized cut algorithm [43] is adopted. To this end, we further define the diagonal matrix  $D$  as  $D_{i,i} = \sum_j W(i, j)$  and an eigen-decomposition problem:

$$D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}}z = \lambda z \quad (19)$$

is solved using the Lanczos algorithm [44]. Since the left-hand-side matrix in equation (19) is symmetric positive definite, all the eigen values are positive real numbers. Moreover, it is noted that the dimension of all eigenvectors is the same the number of points in  $U$ . Among them, the eigenvector corresponding to the second smallest eigenvalue is picked. A standard k-means clustering will decompose  $\mathcal{Q}$  into two subsets, denoted as  $U_1$  and  $U_2$ .

With the point set decomposed,  $U_1$  and  $U_2$  are registered separately to  $\mathcal{P}$ . In particular, we note that the cost function in equation (12) is non-symmetric between  $\mathcal{P}$  and  $\mathcal{Q}$  it only traverses the moving point set. This design naturally provides the capability of registering a partial structure to an entire one, which is the property we expect when registering  $U_1$  and  $U_2$  to  $\mathcal{P}$ . Subsequently, their registration results are evaluated again by equation (15) in order to determine if further decomposition of  $U_1$  and/or  $U_2$  is necessary. This process continues and finally, the decomposition-registration cycle stops when the criteria (user-assigned or automatically computed as a fraction of the diagonal of the bounding box of the point set) is met in all the point subsets. As shown in Theorem 1, convergence is guaranteed.

**Theorem 1 (Convergence):** Given any positive error criteria  $\varepsilon > 0$ , the above hierarchical scheme will find the decomposition  $\{U_i\}$  of  $\mathcal{Q}$  and the corresponding transformations  $\{T_i\}$  such that

$$\frac{1}{|\mathcal{Q}|} \sum_i \sum_{q \in U_i} \|Cl(T_i q) - T_i q\|^2 < \varepsilon. \quad (20)$$

It is noted that the above theorem states the ideal case. In practice, however, the existence of the regularization term in equation (12) will make the theorem work approximately. Finally, The procedure in this section can be summarized as in Algorithm 3.

As shown in Theorem 1, the decomposition can always be performed to achieve arbitrary small registration error. Moreover, we observe that in many situations, the data set can be grouped into a few components, where a single affine transformation is descriptive enough for each group. Hence the nonlinearity of the proposed deformable transformation is



adaptive, and it is not necessary for the decomposition level to be so great that each sub-point set only contains a single

### Algorithm 3

#### Registration & Decomposition ( $\mathcal{P}$ $\mathcal{Q}$ )

---

**Require:** Fixed Point set  $\mathcal{P}$  moving point set  $\mathcal{Q}$

- 1: Register  $\mathcal{Q}$  to  $\mathcal{P}$  as in Algorithm 2
- 2: **if** Error is above criteria **then**
- 3:   Decompose  $\mathcal{Q}$  into  $\mathcal{U}_1 \cup \mathcal{U}_2$
- 4:   **for**  $i = \{1, 2\}$  **do**
- 5:     Registration & Decomposition( $\mathcal{P}$   $\mathcal{U}_i$ )
- 6:   **end for**
- 7: **else**
- 8:   **return** point subsets of  $\mathcal{Q}$  and their registration parameters.
- 9: **end if**

---

point. For instance, in the example shown in Section IV-C, an affine transformation is sufficient for each section of finger between two joints. Therefore, the proposed method provides a nice balance between nonlinearity and optimality.

### C. Fusion of Affine Transformations

In the previous section, the moving point set is decomposed and registered separately to the fixed point set. Although doing so more accurately addresses the local deformation and reduces the registration error at different scale, it generates more than two affine transformations, each of which applying to certain portion of the moving point set. On the other hand, in the final result, what one expects is a coherent transformation applied on the whole moving point set. Hence we need to “stitch” the pieces together.

More generally, any transformation is locally approximated to the first order by an affine transformation. However, in practice it is not only computationally intensive to calculate the local transformation at each point, but also not necessary because in many cases nearby points undergo similar affine transformation. Therefore, the domain of the transformation can be decomposed into several components as discussed in the previous section, each of which is transformed by an affine transformation, and the transformation of the whole domain is the combination of them. However, doing so poses the problem of how to construct the global transformation from each of these components.

More precisely, in the 3D Euclidean space  $\mathbb{R}^3$ , there are  $N$  affine transformations parametrized by:  $(A_i, T_i)$  with  $A_i \in \mathbb{R}^{3 \times 3}$ ,  $i = 1, \dots, N$  and  $T_i \in \mathbb{R}^{3 \times 1}$ ,  $i = 1, \dots, N$ . In addition, each affine transformation corresponds to a weighting function  $\omega_i: \mathbb{R}^3 \rightarrow [0, 1]$ ,  $i = 1, \dots, N$ , where high  $\omega_i(\mathbf{x})$  indicates the  $i$ -th transformation  $(A_i, T_i)$  plays a dominant role in the region closed to  $\mathbf{x}$ . To fuse  $(A_i, T_i)$ 's into a single (nonlinear) transformation  $A$  while respecting their corresponding  $\omega_i$ 's, it was shown in [45] that a linear combination in the Euclidean space given by  $\sum_i \omega_i(\mathbf{x}) (A_i \mathbf{x} + T_i)$  will develop singularities.

To address this issue, form the affine transformations a Lie group, and the “infinitesimal” transformations associated with them defines the corresponding Lie algebra [45]. Consequently, instead of performing the above straightforward weighted combination of the

affine transformations in the Lie group, one seeks the alternative in the Lie algebra, and then the

#### Algorithm 4

##### Fusion of Local Affine Transformations to Form a Diffeomorphism

---

**Require:** A sequence of point subsets:  $\{U_i \subset \mathcal{Q}: i = 1, \dots, K\}$  of the moving point set  $\mathcal{Q}$

**Require:** A set of affine transformations:  $\{B_i \in \mathbb{R}^{4 \times 4}: i = 1, \dots, K\}$ , corresponding to each  $U_i$  above.

```

1:  for Each  $\{U_i \subset \mathcal{Q}: i = 1, \dots, K\}$  do
2:    Compute matrix logarithm  $L_i$  or  $B_i$ 
3:  end for
4:  for Each  $\mathbf{x} \in \mathcal{Q}$  do
5:    for Each  $\{U_i \subset \mathcal{Q}: i = 1, \dots, K\}$  do
6:      Compute weights  $\omega_i(\mathbf{x})$  using equation (21)
7:    end for
8:    Compute weighted sum in Lie Algebra using equation (22)
9:    Compute final transformation  $B(\mathbf{x})$ , where  $B$  is the matrix exponential of  $L$  in equation (22).
10:  end for

```

---

weighted average is mapped back as the final transformation. It is shown in [45] that such a transformation is a diffeomorphism. More explicitly, we assume the hierarchical decomposition registration scheme results in a list of point subsets:  $\{U_i \subset \mathcal{Q}: i = 1, \dots, K\}$  of the moving point set  $\mathcal{Q}$  and their corresponding affine transformations (in homogeneous coordinate):  $\{B_i \in \mathbb{R}^{4 \times 4}: i = 1, \dots, K\}$ . To fuse them, a weighting function  $\omega_i: \mathbb{R}^3 \rightarrow [0, 1]$  is designed for each transformation  $B_i$  (and the corresponding  $U_i$ ) as  $\omega_i(\mathbf{x}) = \omega_i(\mathbf{x}) / \sum_j \omega_j(\mathbf{x})$  with

$$\tilde{\omega}_i(\mathbf{x}) = e^{-\|\mathbf{x} - Cl_i(\mathbf{x})\|^2} \quad (21)$$

where the function  $Cl_i: \mathbb{R}^3 \rightarrow U_i$  returns the closest point in the point set  $U_i$  to a given position  $\mathbf{x} \in \mathbb{R}^3$ . It can be seen that  $\omega_i$  indicates each transformation  $B_i$ 's spatial extension: the weight of the transformation  $B_i$  is high around the point set  $U_i$  and is low at where far from  $U_i$ . Referring to the above discussion, the transformation matrix  $B_i$  is in the Lie group (whose direct weighted sum,  $\sum_i \omega_i B_i$ , develops singularities). Next, the infinitesimal motion  $L_i$  in the Lie algebra is computed by the matrix logarithm. Then, the weighting function  $\omega(\cdot)$  is evaluated at  $\mathbf{x}$  and the weighted average infinitesimal transformation in the Lie algebra is obtained as

$$L(\mathbf{x}) := \sum_i \omega_i(\mathbf{x}) L_i. \quad (22)$$

Finally, the matrix exponential  $B(\mathbf{x})$  of  $L(\mathbf{x})$  is computed and the image of  $\mathbf{x}$  is  $B(\mathbf{x}) \cdot \mathbf{x}$ . As shown in [45], the transformation  $B(\mathbf{x}): \mathbf{x} \in \mathbb{R}^3 \times \{1\}$  is a diffeomorphism. The construction of the diffeomorphism is summarized in Algorithm 4.

## IV. Experiments

In this section, we demonstrate the use of PF to solve an optimization problem. In particular, we validate the claim in Section II-B that the filtering based on evolving process model [equation (9)] outperforms filtering based on identity process model [equation (7)], in a

general optimization problem. Then, various qualitative and quantitative examples are provided to show the performance and robustness of the proposed registration method.

### A. Optimization via PF, Comparison of Identity/Evolving Process Models

In this experiment, we demonstrate the capability of particle filtering for solving the optimization problems. Furthermore, we compare the efficiency of PF on a system with the identity process model as in equation (7) with one that uses the evolving process model in equation (9).

To this end, the objective cost function  $C : \mathbb{R}^2 \rightarrow \mathbb{R}$  is constructed as:

$$C(x, y) = 20 + x^2 + y^2 - 10(\cos(0.8\pi x) + \cos(0.8\pi y)) \quad (23)$$

whose graph is shown in figure 1.

In order to optimize this cost function, we design a dynamical system with identity process model by letting  $\mathbf{u}$  in equation (7) be zero mean Gaussian distributed with standard deviation of 0.5, and  $\mathbf{v}$  in equation (8) to be negative exponential distribution with rate parameter as 0.1. Furthermore, the initial distribution of the state variable is assumed to be uniformly distributed on  $[-6, 6] \times [-6, 6]$ . In Figure 2, we show the 50 evolving particles for optimizing the cost function (23). At convergence, the minimizer in the current particles are picked as the optimization result. It is noted that for a 2D problem like this, 50 particles are more than enough. However, in order to demonstrate the evolution of the particles more clearly, we used more than a sufficient number of particles.

Although the final particles cluster about the global optimizer, we expect them to be more concentrated so that the final estimate for the optimizer will be more accurate. However, reducing the variance of the process or observation noise is not a good choice since that will cause particle impoverishment, which is of course undesired [41]. On the other hand, by using the evolving process model as in equation (9), the particles evolve towards the (local) minima in the process, instead of random walks, which increases both the accuracy and the convergence rate. Indeed, we use the three steps of the BFGS optimization as described in Algorithm 2. The particle filtering with this evolving process model is used. In order to make a fair comparison, all the random numbers used for Figure 2 are reused. Therefore, the prior particles are the same as Figure 2(a). The particles after the 3, 6, 9-th iterations are shown in Figure 3.

It can be observed from Figure 3 that the particles are much more concentrated at the global optimizer with even fewer particle filtering iterations. Although the deterministic optimization in the process model makes each iteration longer, nevertheless, the total running time is only half of the previous experiment.

### B. Affine ICP Using Particle Filtering (Affine-PF-ICP)

Following the previous experiment where the PF is applied in a general optimization problem, here we use the affine point set registration scenario to demonstrate that the affine-PF-ICP in Section III-A successfully overcomes the local minima problem. Specifically, we employ an example where the ground truth is known *a priori*. In Figure 4(a), the blue point set is a 3D point set sampled from human brain white matter. (While the generation of this point set is detailed in Section IV-E, such details are not necessary for the understanding of the present experiment.) This blue point set is transformed into another affine pose as shown by the red one, and the objective is to register the red point set to the blue one.

For this purpose, minimizing the affine-ICP cost function defined in equation (12) gives the result shown in Figure 4(b). The blue points are again from the fixed point set while the red ones are from the registered moving point set. Due to the local minima, the poses of the two point sets are not well registered. Contrastingly, the affine-PF-ICP correctly restores the original pose as shown in Figure 4(c).

To further analyze the performance of the affine-PF-ICP quantitatively, the following experiments were conducted: The blue point set in Figure 4(a) is again fixed, and a copy of it is perturbed by a random affine transformation and is then employed as the moving point set. Three types of registration are compared, namely affine-ICP, affine-CPD [26], as well as our proposed method. All three methods are run until convergence.

In order to evaluate the performances, the registration error (RegE) is measured for the initial perturbation as well as after convergence. As a preprocessing step of the registration, the center (algebraic mean of all the points) of point sets are aligned. Therefore, only the perturbation of affine matrix is measured. Four trials are performed: in the  $i$ -th trial, the fixed point set is perturbed by an initial affine matrix, which is the sum of an identity matrix with a random matrix whose elements are normally distributed with zero mean and standard deviation of  $0.1i$ ;  $i = 1, 2, 3, 4$ . In each trial, 1000 realizations are simulated. The RegE after such initial transformation is measured and plotted in Figure 5(a), and the RegEs after applying different registration methods are plotted in Figure 5(b) through Figure 5(f). Due to the fact that the point-wise correspondence is not known for the algorithms, and the energy functionals to be optimized are not RegE, therefore, in each trial, there are cases whose final RegEs are larger than the initial RegEs. In these cases, the point sets are transformed to some local minima of their respective energy functional, and result in larger final RegE. Moreover, as can be observed in Figure 5(b), the affine-ICP is very sensitive to the perturbation. However, by performing the affine-ICP in the particle filtering framework, with the increase of the number of particles, the robustness of the registration improves substantially [Figure 5(d) through 5(f)]. As a comparison, the affine-CPD also performs very well except for a few cases where the perturbation is really large [Figure 5(c)].

Furthermore, we therefore performed the noise test on the two data sets: in the bounding box of the point set, we added uniformly distributed points. In total, 10 trials were performed. In the  $i$ -th trial, the number of added points is  $i \times N/10$  where  $N$  is the number of points in the original point set. In each trial, 1000 simulations were done. For each simulation, the fixed point set is transformed by a random affine matrix, formed by adding an identity matrix with a random matrix whose elements are normally distributed with zero mean and standard deviation of 0.4. The final RegE are plotted in Figure 6 of this document.

### C. Walk-Through Example

After illustrating the (linear) affine-PF-ICP, we use this example to demonstrate the whole process of the (nonlinear) hierarchical diffeomorphic registration algorithm. Figure 7 shows two sets of points. (For better appreciation of their relative positions, we also show the surface. However, only the points are used in the algorithm.) The fixed point set is colored in cyan and it is extracted from the public available data set #72 on [shapes.aimatshape.net](http://shapes.aimatshape.net).<sup>1</sup> The moving point set, in yellow, is number #759.<sup>2</sup> Both point sets are the index and middle fingers from different left hands. It can also be observed that apart from the affine pose difference between the two point sets, there are local deformations, e.g., the fingers of #72 are bent, while those of #759 are straighter.

<sup>1</sup><http://shapes.aimatshape.net/view.php?id=72>.

<sup>2</sup><http://shapes.aimatshape.net/viewgroup.php?id=759>.

The result of applying the affine-PF-ICP is shown in Figure 8 where the fixed point set is in cyan and the registered moving point set is in yellow. It can be observed that a single affine transformation is not enough to align the two point sets. In fact, visual inspection reveals that the angles between the two fingers in the two point sets are different, as well as there are finer deformations. Applying the decomposition scheme in Section III-B, the registered moving point set is separated into two, as shown in Figure 9(a).

Each point subset is then registered to the fixed point set separately. Due to the asymmetry in the registration cost function defined in equation (12), registering a partial structure to the whole structure is naturally handled. Indeed, comparing Figure 9(b) to Figure 8, we see the angular difference problem is resolved. However, not surprisingly it also results in tearing, which is highlighted in Figure 9(c) and 9(d). The moving point set is further decomposed hierarchically. After registration at a finer scale, the computed transformations are fused into a single diffeomorphic registration, shown in Figure 10.

As can be observed in enclosed region highlighted by the green circle of Figure 10(d), the multiple transformations are nicely “stitched” together at the regions highlighted by the green ellipse, such that there is no tearing as occurred in Figure 9(d) and the components are smoothly connected. Moreover, the sub-point sets are better registered (Figure 10(a)10(b)) than at coarser levels. Interestingly, in this example the decomposition is consistent with the anatomical structure of the fingers. However, in general the purpose of decomposition is for finer scale registration and does not necessarily correspond to anatomical decomposition.

Quantitatively, we plot the reduction of the registration error in Figure 11. The vertical axis is the error defined in equation (15) (The fixed point set has the range in:  $[-0.26, 0.13] \times [0.26, 0.85] \times [-0.24, 0.31]$ .) In the figure the dashed line with the downward triangle marker plots the log of error of just affine-PF-ICP without decomposition. It can be seen that the error stops decreasing after a certain number of iterations. However, with the decomposition, the error could be further reduced by registration at finer level, shown as the “square” line. After the second decomposition, the error further drops, as indicated by the “upper-triangle” line, and the “star” line after the third decomposition. Furthermore, it is noted that when there are several components, the largest error is plotted. Although the current result after three decomposition is satisfying, further decomposition can be carried on if desired. The total running time for this two data sets is 24 seconds on a Intel i7 2.8GHz Quad Core machine.

#### D. Left-Right Hand Registration

While the previous experiment demonstrated the steps of the algorithm, here we apply the algorithm to the entire hand data sets. Furthermore, in this experiment, we demonstrate the capability of the proposed method in handling the registration problem when the two point sets are mirrored. Specifically, the fixed point set is the entire *right* hand point set public available at <http://shapes.aimatshape.net/view.php?id=72>, and the moving point set is a point set of a *left* hand at <http://shapes.aimatshape.net/view.php?id=355>. The fixed and the moving point sets are shown in cyan and yellow, respectively, in Figure 12. It is noted that the locations and the ranges of the original two point sets differ substantially. Therefore in this experiment, as a preprocessing step, the centers of mass for the two sets are shifted to the origin and the largest dimensions (vertical direction in Figure 12) of the point sets are normalized to  $[-0.5, 0.5]$  before applying our registration scheme. We apply the proposed method, in which the number of particles is set to 300, and the convergence criteria is set to  $10^{-3}$ . The algorithm hierarchically divides the moving hand point sets into 14 clusters, and we obtain the results as shown in Figure 13. Moreover, it is noted that the different colors are used to contrast two adjacent regions. Therefore, two non-adjacent regions may have the same color but still represent two different clusters.

As a comparison, the non-rigid CPD algorithm [26] is applied to register the two point sets. In order for the algorithm to perform well, we first flipped the horizontal-direction so as to make the moving point set also a left-hand. Furthermore, we applied the CPD-affine registration which is followed by a CPD-non-rigid registration. The registration results are evaluated by measuring the RegE and the values are given in Table I. The total running time for this two data sets is 72 seconds on a Intel i7 2.8GHz Quad Core machine.

### E. Tract Registration

In this set of experiments we further test the algorithm on larger and more complex point sets. Here each point set represents the neuro-tracts of the human white matter. Different functional regions in the cerebral cortex (gray matter) are connected by those tracts. Moreover, clinical correlations have been found between the shape of the tracts and certain brain diseases such as schizophrenia [46]. Hence, under the scenario of statistical analysis of fiber tracts of different subjects, it is crucial to register them into a common coordinate system. Furthermore, the tracts can be categorized into different groups, each of which contains tracts connecting similar functional regions of the gray matter. However, on the one hand it is very time consuming for a neuroscientist to hand group the tracts, and on the other hand the fully automatic tract classification is a challenging problem in its own right. Thus, in this experiment one set of fiber tracts has been grouped into several groups manually and is used as the registration template, to which all the others are registered to and clustered accordingly. The diffusion weighted images are collected at the Brigham and Women's Hospital and the tracts (represented by points) are obtained by the two-tensor tractography method [39].

Figure 14 shows the original pose and position of the clustered template fiber tracts (point set), and one moving point set (un-clustered). It can be seen that in addition to the size, the two point sets are flipped, which makes the previous local-optimization based algorithms not well-suited to the registration task at hand.

The subplots of Figure 15 show the results of registering the moving point set to the template, using the affine-PF-ICP method. Since this figure only shows the performance of the registration, for clarity of display, the template point set is plotted in cyan and the registered moving point set is in yellow. It can be seen that the fine part (noted in the box) is not well captured.

Next, the proposed algorithm is applied. In the experiments, it was observed that the decomposition takes a large portion of the total running time. In order to reduce the time, the point set to be decomposed is sub-sampled before clustering. More explicitly, the number of points in the point set is iteratively divided by 10 so that the resulting number is below  $10^4$ . With the size determined, the samples are extracted uniformly from the original point set. The decomposition process in Section III-B is then executed on this sub-set. Afterwards, the classes of the other points are determined by that of the closest point that is in the decomposed sub-set. Moreover, the decomposition criteria set at  $10^{-2}$ . Finally, the registration result for one of the four data sets is shown in Figure 16.

However, note that the clusters obtained by the proposed method shown on the right column of the Figure 16 are just for the purpose of filtering of the diffeomorphism, and they do not have anatomical significance. In fact, utilizing the fact that the template point set has been clustered into several anatomically significant groups, we are able to achieve diffeomorphic registration as well as obtain the anatomically significant groups for the new fiber tracts. In fact, after the particle filtering affine registration on the entire point set, the moving point set is assigned into groups using the closest point criterion. That is, for each point in the affine-PF-ICP registered moving point set, its class is decided by that of the closest point in the



template point set. Therefore, each moving point subset is registered to its corresponding template subset by the method described in Section III-A.

In this test, the template point set has 908894 points and has been clustered into 10 tract bundles, each of which is registered to its corresponding bundle in the fixed point set. Based on the new registration result, the moving point set is re-grouped to achieve a better result. The grouping/registration cycles are iterated and finally, the 10 paired point subset registration results are fused to form a global diffeomorphic transformation, which is shown in Figure 17. The results obtained here are very close to the one in Figure 16, except that the clustering of the moving points now is according to the anatomical structure.

Indeed, the registration scheme simultaneously provides a clustering for the moving point set. Using different colors for different clusters, they are shown in Figure 18.

The method is further tested on more tracts data sets. Each point set has a similar initial position and pose configuration as detailed above. The results are given in table II. It can be seen that the proposed method, with or without using the clustering information for the fixed point set, achieves similar results. This is because the number of the manual clusters, 10, is similar to that generated by decomposition, which is 16 given the decomposition criteria at  $10^{-2}$ .

The registration performance is achieved by a high numbers of degrees of freedom (DOF) in the transformation. Indeed, for 10 or 16 clusters, there are in total 120 or 160 DOFs. However, although the overall number of DOFs is high, in each and every optimization and filtering step, the problem scale is always 12. This is the main reason that the proposed method achieves nonlinear registration in the filtering framework within feasible time. In contrast, for more standard non-rigid transformations schemes, the number of optimization variables often exceeds 1000 (for 10 spline control nodes in each dimension), or even more where the transformation is defined in a pointwise manner, e.g., via a vector field. Therefore, the proposed method achieves filtering for a diffeomorphism through filtering in the affine group. In addition, the proposed algorithm has the advantage that the nonlinearity of the registration transformation is refined adaptively only where a finer resolution is necessary. This substantially increases the accuracy as well as keeps the computation load at an acceptable level. Moreover, no matter how high fine resolution the transformation reaches, the computation of optimization and filtering is always constrained in the 12 dimensional space.

## V. Conclusion

In this work we proposed a multi-scale registration method to perform filtering in the infinite dimensional diffeomorphism group. First, by treating the point set affine registration problem as one of parameter estimation, we could employ particle filtering to achieve a more global optimal solution. Second, to further address the local deformation between the point sets, the affinely registered moving point set was decomposed into two subsets in a six dimensional feature space, where the moving point coordinates were augmented by their error vector. Then, the subsets were registered to the fixed point set separately in order to correct the disparity at a finer level. Finally, the local transformations were fused together in order to form a smooth and invertible nonlinear transformation. The method was tested on point sets with prominent local deformation (point sets of human fingers), as well as complex local structures (brain fiber tracts), in order to show the performance and robustness of the registration results both qualitatively and quantitatively.

Further work includes extending the current registration framework to work with groups of data sets, to replace the arbitrary template point set selection. Moreover, we will extend the

point set registration to open curve registration which may be more suited for the fiber tract registration and clustering tasks. In such scenarios, the decomposition of curves will be performed in the more complex space of spatial curves.

## Acknowledgments

The authors would like to thank the anonymous reviewers for their help in improving the manuscript. The authors would also thank Dr. J. Malcolm and Dr. R. Sandhu for their help in preparing the tracts data sets. Information on the National Centers for Biomedical Computing can be obtained from <http://nihroadmap.nih.gov/bioinformatics>.

This work was supported in part by grants from the NSF, AFOSR, and ARO, the NIH under Grant NAC P41 RR-13218 through Brigham and Women's Hospital, and under Grant R01 MH82918, and the National Alliance for Medical Image Computing, funded by the National Institutes of Health through the NIH Roadmap for Medical Research under Grant U54 EB005149. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Birsen Yazici.

## Appendix A

### Proof of the Theorem 1

*Proof:* The proof consists of two parts:

First we show the average error magnitude decreases monotonically at every scale. Indeed, assume the optimal affine transformation  $B$  is computed by any registration algorithm, which aligns the point set  $\mathcal{Q}$  to  $\mathcal{P}$  i.e.

$$B = \arg \min_B \frac{1}{N} \sum_{q \in \mathcal{Q}} \|Cl(Bq) - Bq\|^2. \quad (24)$$

Furthermore,  $\mathcal{Q}$  is decomposed into  $U_1$  and  $U_2$ , and each of them is registered to  $\mathcal{P}$  starting from  $B$ . Then by the optimization process discussed in Section III-A, it is guaranteed that the resulting two affine transformations  $B_1$  and  $B_2$  will yield a smaller error:

$$\sum_{q \in U_1} \|Cl(B_1q) - B_1q\|^2 + \sum_{q \in U_2} \|Cl(B_2q) - B_2q\|^2 \leq \sum_{q \in \mathcal{Q}} \|Cl(Bq) - Bq\|^2. \quad (25)$$

Indeed, even without further optimization at a finer level, i.e.,  $B_1 = B_2 = B$ , the equality in equation (25) already holds. Furthermore, since the optimization at finer level monotonically decreases the cost functional, the non-strict inequality in equation (25) always holds.

Second, we claim the average error magnitude approaches zero in finite number of steps. In fact, as the decomposition-registration proceeds, because both point sets are finite, the following limit case can always be reached in finite time:  $\mathcal{Q}$  is decomposed into  $N$  sub-point-sets, each of which contains less than 12 points. In such a case, determining an affine transformation between them is an under-determined problem and can be solved with zero error.

## References

1. Besl P, McKay H. A method for registration of 3-D shapes. IEEE Trans. Pattern Anal. Mach. Intell. 1992 Feb.vol. 14(no. 2):239–256.
2. Chen Y, Medioni G. Object modeling by registration of multiple range images. Image Vis. Comput. 1992; vol. 10(no. 3):145–155.
3. Eggert D, Lorusso A, Fisher R. Estimating 3-D rigid body transformations: A comparison of four major algorithms. Mach. Vis. Appl. 1997; vol. 9(no. 5):272–290.

4. Rusinkiewicz, S.; Levoy, M. Efficient variants of the ICP algorithm; Proc. 3rd Int. Conf. 3-D Digital Imag. Model; 2001. p. 142-151.
5. Stewart C. Robust parameter estimation in computer vision. SIAM Rev. 1999; vol. 41(no. 3):513–537.
6. Horn B. Closed-form solution of absolute orientation using unit quaternions. J. Opt. Soc. Amer. A. 1987; vol. 4(no. 4):629–642.
7. Dryden, I.; Mardia, K. Statistical Shape Analysis. New York: Wiley; 1998.
8. Fitzgibbon A. Robust registration of 2D and 3D point sets. Image Visi. Comput. 2003; vol. 21(nos. 13–14):1145–1153.
9. Jiang J, Cheng J, Chen X. Registration for 3-D point cloud using angular-invariant feature. Neurocomputing. 2009; vol. 72(nos. 16–18):3839–3844.
10. Sharp G, Lee S, Wehe D. ICP registration using invariant features. IEEE Trans. Pattern Anal. Mach. Intell. 2002 Jan.vol. 24(no. 1):90–102.
11. Chui H, Rangarajan A. A new point matching algorithm for non-rigid registration. Comput. Vis. Image Understand. 2003; vol. 89(nos. 2–3):114–141.
12. Gold S, Rangarajan A, Lu C, Pappu S, Mjolsness E. New algorithms for 2D and 3D point matching pose estimation and correspondence. Pattern Recognit. 1998; vol. 31(no. 8):1019–1031.
13. Li, H.; Hartley, R. The 3D-3D registration problem revisited; Proc. IEEE Int. Conf. Comput. Vis; 2007 Oct.. p. 1-8.
14. Bookstein FL. Four metrics for image variation. Proc. 11th Int. Conf. Inf. Process. Med. Imag., Progr. Clin. Biol. Res. 1991; vol. 363:227–240.
15. Christensen G, Joshi S, Miller M. Volumetric transformation of brain anatomy. IEEE Trans. Med. Imag. 1997 Dec.vol. 16(no. 6):864–877.
16. Rohr K, Stiehl H, Sprengel R, Buzug T, Weese J, Kuhn M. Landmark-based elastic registration using approximating thin-plate splines. IEEE Trans. Med. Imag. 2001 Jun.vol. 20(no. 6):526–534.
17. Chua C, Jarvis R. 3D free-form surface registration and object recognition. Int. J. Comput. Vis. 1996; vol. 17(no. 1):77–99.
18. Gelfand, N.; Mitra, N.; Guibas, L.; Pottmann, H. Robust global registration; Proc. 3rd Eurograph. Symp. Geometry Process; 2005. p. 197-186.
19. Huber D, Hebert M. Fully automatic registration of multiple 3D data sets. Image Vis. Comput. 2003; vol. 21(no. 7):637–650.
20. Johnson A. Surface landmark selection and matching in natural terrain. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2000 Jun.vol. 2:413–420.
21. Johnson A, Hebert M. Using spin images for efficient object recognition in cluttered 3D scenes. IEEE Trans. Pattern Anal. Mach. Intell. 1999 May; vol. 21(no. 5):433–449.
22. Tsin, Y.; Kanade, T. A correlation-based approach to robust point set registration; Proc. Eur. Conf. Comput. Vis; 2004. p. 558-569.
23. Jian B, Vemuri B. A robust algorithm for point set registration using mixture of Gaussians. Proc. IEEE Int. Conf. Comput. Vis. 2005 Oct.vol. 2:1246–1251. [PubMed: 19169422]
24. Wang F, Vemuri BC, Rangarajan A, Eisenschenk SJ. Simultaneous nonrigid registration of multiple point sets and atlas construction. IEEE Trans. Pattern Anal. Mach. Intell. 2008 Nov.vol. 30(no. 11):2011–2022. [PubMed: 18787248]
25. Granger, S.; Pennec, X. Multi-scale EM-ICP: A fast and robust approach for surface registration; Proc. 7th Eur. Conf. Comput. Vis., IV; 2002. p. 418-432.
26. Myronenko A, Song X. Point-set registration: Coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. 2010 Dec.vol. 32(no. 12):2262–2275. [PubMed: 20975122]
27. Glaunes J, Trouvé A, Younes L. Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. Proc. IEEE Comput. Vis. Pattern Recognit. 2004 Jul.vol. 2.:712–718.
28. Lakaemper R, Sobel M. Correspondences between parts of shapes with particle filters. Proc. IEEE Comput. Vis. Pattern Recognit. 2008 Jun.:1–8.
29. Ma B, Ellis R. Unified point selection and surface-based registration using a particle filter. Proc. 8th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. 2005; vol. 8:75–82.

30. Ma, B.; Ellis, Y. Surface-based registration with a particle filter; Proc. 8th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent; 2004. p. 566-573.
31. Sandhu R, Dambreville S, Tannenbaum A. Point set registration via particle filtering and stochastic dynamics. *IEEE Trans. Pattern Anal. Mach. Intell.* 2010 Aug.vol. 32(no. 8):1459–1473. [PubMed: 20558877]
32. Moghari M, Abolmaesumi P. Point-based rigid-body registration using an unscented Kalman filter. *IEEE Trans. Med. Imag.* 2007 Dec.vol. 26(no. 12):1708–1728.
33. Ziyen U, Sabuncu M, Grimson W, Westin C. Consistency clustering: A robust algorithm for group-wise registration, segmentation and automatic atlas construction in diffusion MRI. *Int. J. Comput. Vis.* 2009; vol. 85(no. 3):279–290. [PubMed: 20442792]
34. Pitiot A, Bardin E, Thompson P, Malandain G. Piecewise affine registration of biological images for volume reconstruction. *Med. Image Anal.* 2006; vol. 10(no. 3):465–483. [PubMed: 15963755]
35. Nocedal, J.; Wright, S. Numerical Optimization. New York: Springer-Verlag; 1999.
36. Wolpert D, Macready W. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* 1997 Apr.vol. 1(no. 1):67–82.
37. Gordon N, Salmond D, Smith A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F, Radar Signal Process.* 1993 Apr.vol. 140(no. 2):107–113.
38. Kirkpatrick S, Gelatt C Jr, Vecchi M. Optimization by simulated annealing. *Science.* 1983 May; vol. 220(no. 4598):671–680. [PubMed: 17813860]
39. Malcolm JG, Shenton ME, Rath Y. Neural tractography using an unscented Kalman filter. *Proc. Inf. Process. Med. Imag.* 2009:126–138.
40. Kennel M. KDTree 2: Fortran 95 and C++ Software to Efficiently Search for Near Neighbors in a Multi-Dimensional Euclidean Space. 2004 [Online]. Available: <http://arxiv.org/abs/physics/0408067>.
41. Doucet, A.; De Freitas, N.; Gordon, N. Sequential Monte Carlo Methods in Practice. New York: Springer-Verlag; 2001.
42. Fischler M, Bolles R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM.* 1981; vol. 24(no. 6):381–395.
43. Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2000 Aug.vol. 22(no. 8):888–905.
44. Calvetti D, Reichel L, Sorensen D. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electron. Trans. Numer. Anal.* 1994; vol. 2(no. 1):1–21.
45. Arsigny V, Commowick O, Ayache N, Pennec X. A fast and log-Euclidean polyaffine framework for locally linear registration. *J. Math. Imag. Vis.* 2009; vol. 33(no. 2):222–238.
46. Kubicki M, Park H, Westin C, Nestor P, Mulkern R, Maier S, Niznikiewicz M, Connor E, Levitt J, Frumin M. DTI and MTR abnormalities in schizophrenia: Analysis of white matter integrity. *Neuroimage.* 2005; vol. 26(no. 4):1109–1118. [PubMed: 15878290]

## Biographies



**Yi Gao** received the Ph.D. degree from the Georgia Institute of Technology, Atlanta.

He is currently a Research Fellow with the Psychiatry Neuroimaging Laboratory, Harvard Medical School, Boston, MA. His current research interests include shape and image analyses.



**Yogesh Rath** (M'05) is an Assistant Professor with Brigham and Women's Hospital, Harvard Medical School, Boston, MA. His current research interests include compressed sensing, computer visions, and image processing.

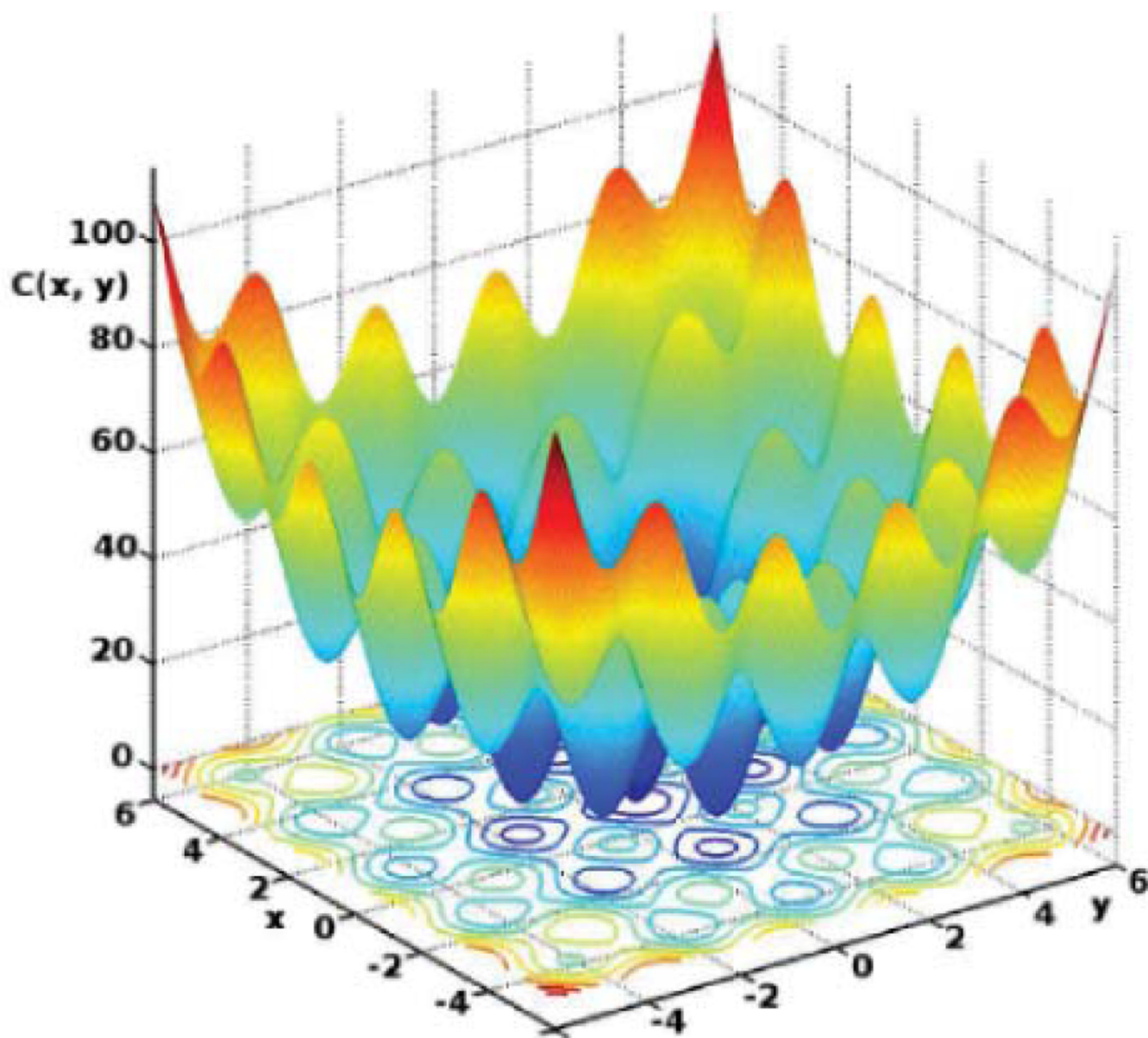


**Sylvain Bouix** (M'03) received the B.Eng. degree from the Institut Polytechnique de Sevenans, Belfort, France, the M.Sc. degree from the University of Kansas, Lawrence, and the Ph.D. degree from McGill University, Montreal, QC, Canada.

He is an Assistant Professor and an Associate Director with the Department of Psychiatry, Psychiatry Neuroimaging Laboratory, Brigham and Women's Hospital, Harvard Medical School, Boston, MA. He is engaged as a mediator between computer scientists and neuroscientists to help improve computer tools for neuroimaging studies. His current research interests include shape analysis of anatomical structures and evaluation of medical imaging techniques.

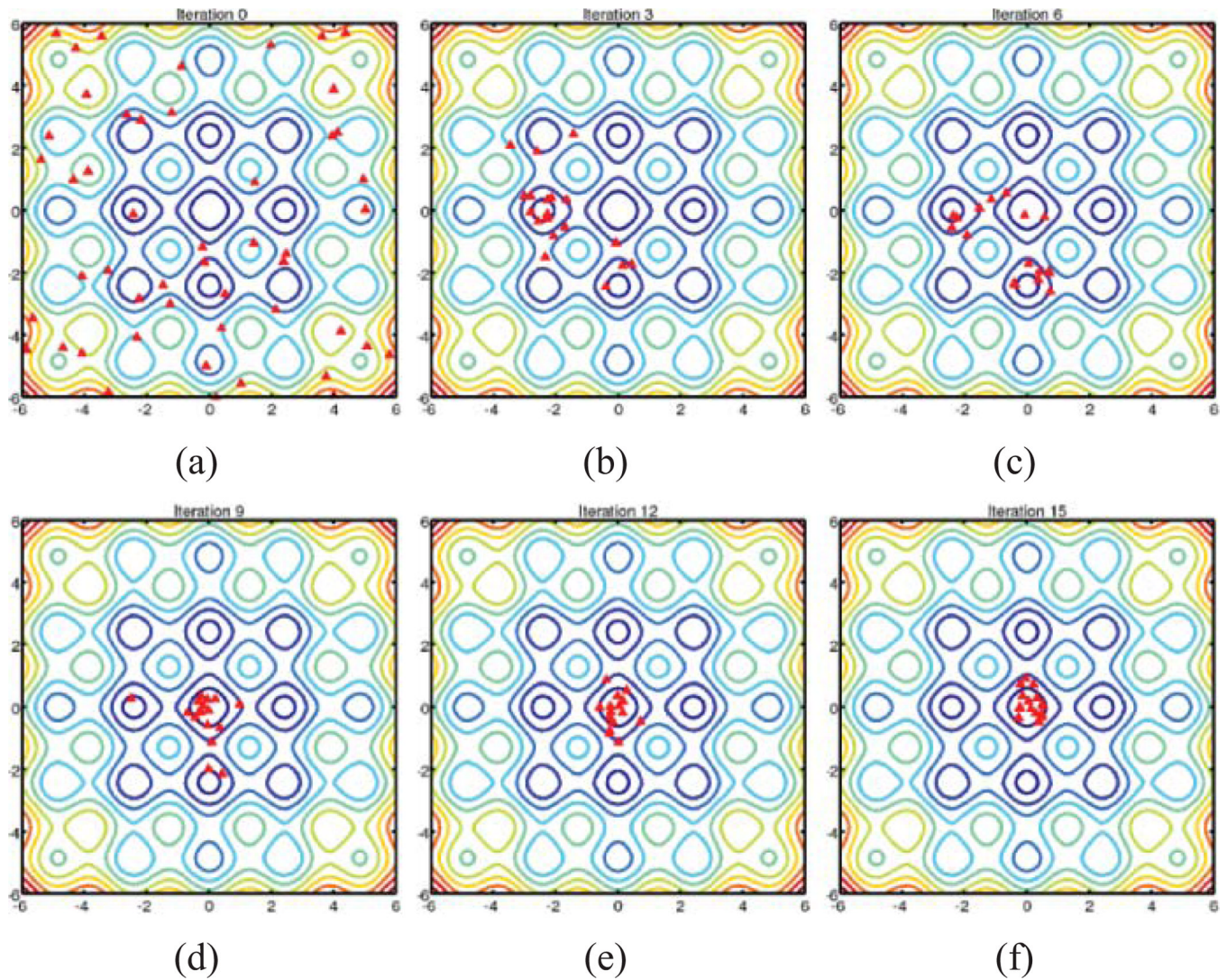


**Allen Tannenbaum** (F'09) is a Faculty Member with the School of Electrical and Computer Engineering, Boston University, Boston, MA. His current research interests include systems and control, computer visions, and image processing.

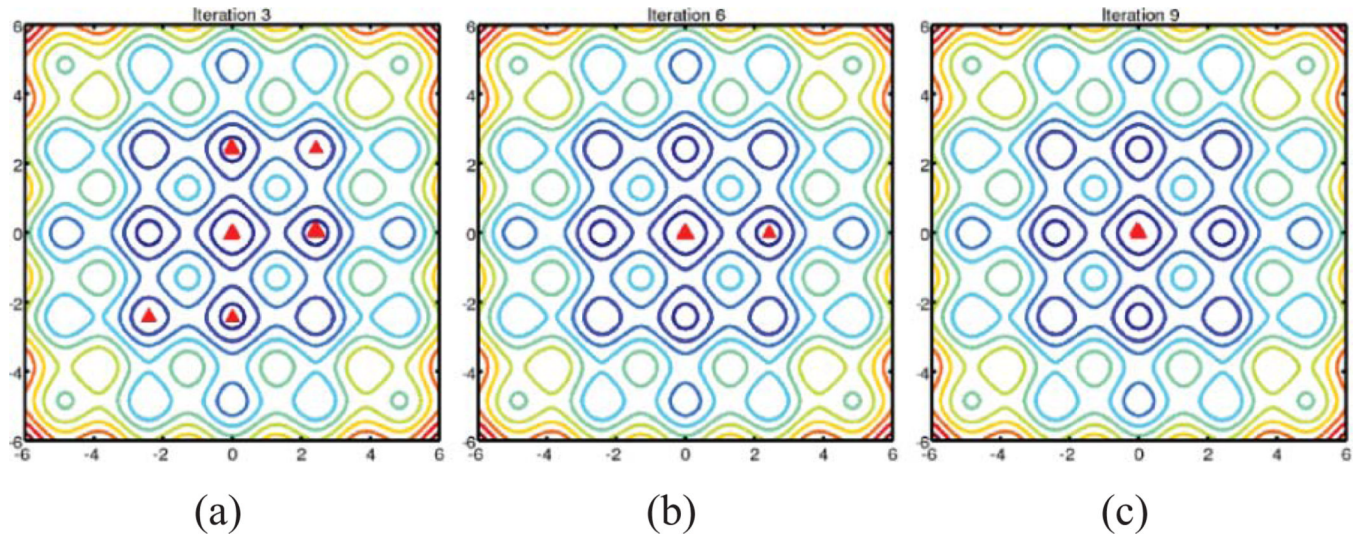


**Fig. 1.**  
Cost function for optimization.

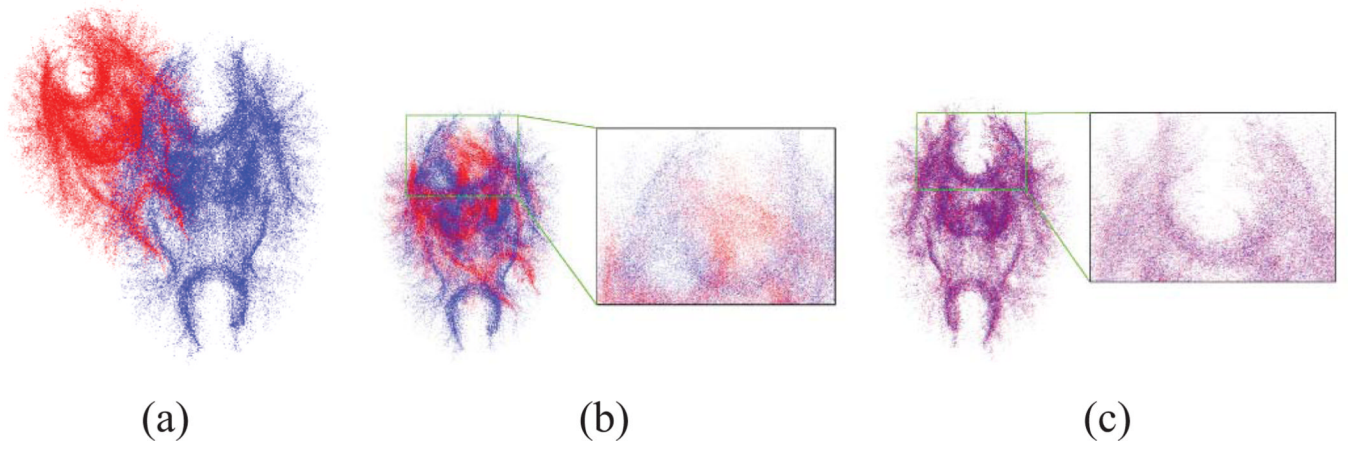


**Fig. 2.**

(a)–(f) 50 evolving particles for optimizing the cost function (23). Red triangles: Particles after the 0th, 3rd, 6th, 9th, 12th, and 15th iterations. Starting from the samples/particles for the prior distribution, the particles first concentrated around local minima and finally converged to the global minimum located at the center of the domain.

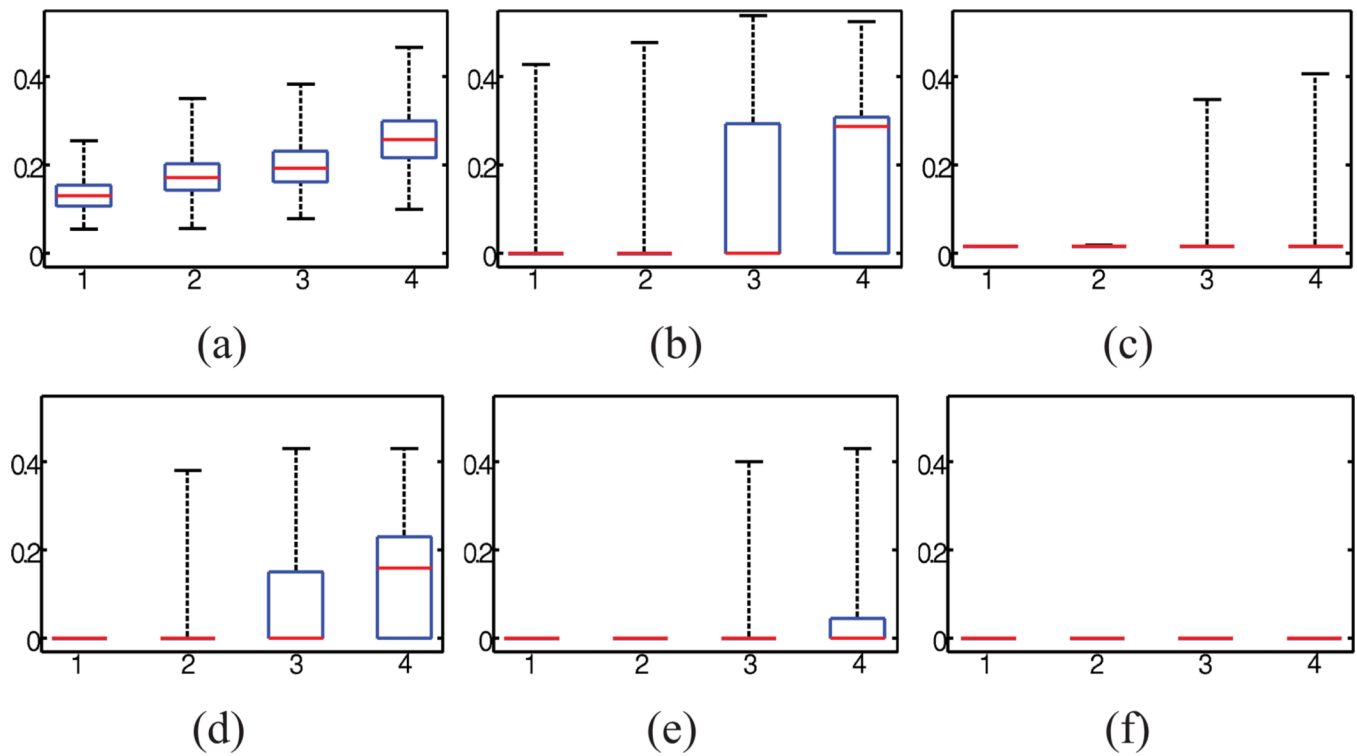
**Fig. 3.**

(a)–(c) 50 evolving particles for optimizing the cost function (23) using the evolving process model in (9). Red triangles: Particles after the 3rd, 6th, and 9th iterations.

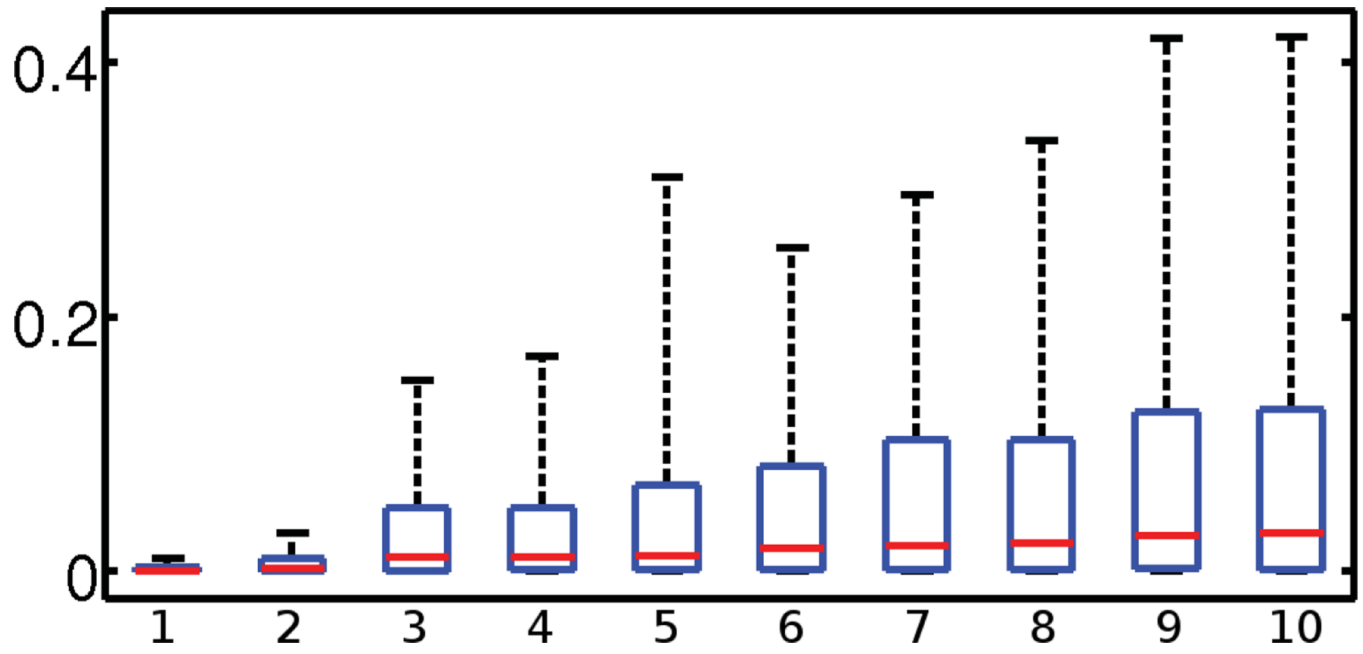
**Fig. 4.**

Registration initial condition and registration results. (a) One point set in two affine poses. (b) Using affine ICP. (c) Using affine-PF-ICP.

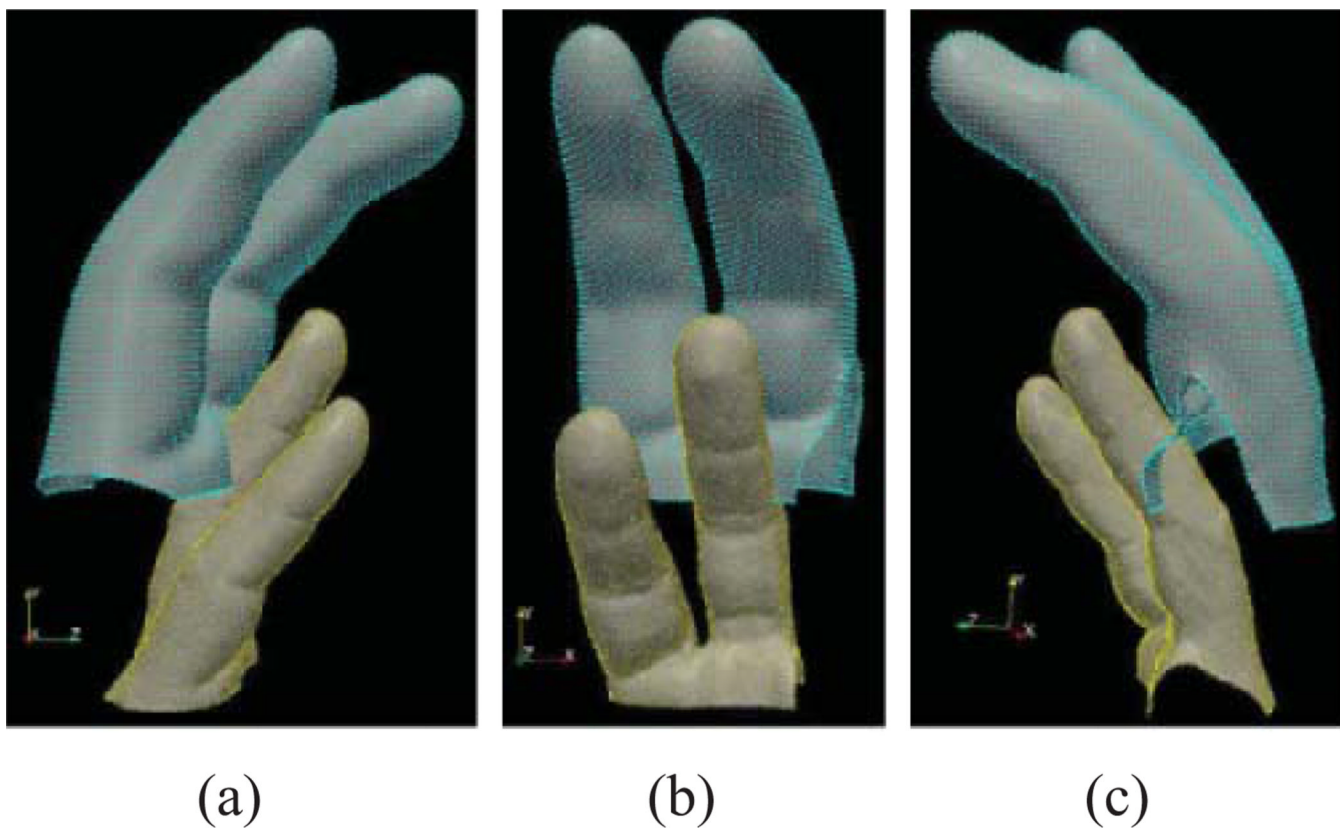


**Fig. 5.**

RegE before and after registration. The horizontal axis is the trial number. In the  $i$ th trial, the initial affine matrix is an identity matrix plus a random matrix whose elements are normally distributed with zero means and a standard deviation of  $0.1 \times i$ . (a) RegE after initial perturbation. (b) RegE after using affine-ICP. (c) RegE after affine-CPD. (d)–(f) RegE after affine-PF-ICP with 30, 100, and 300 particles, respectively.

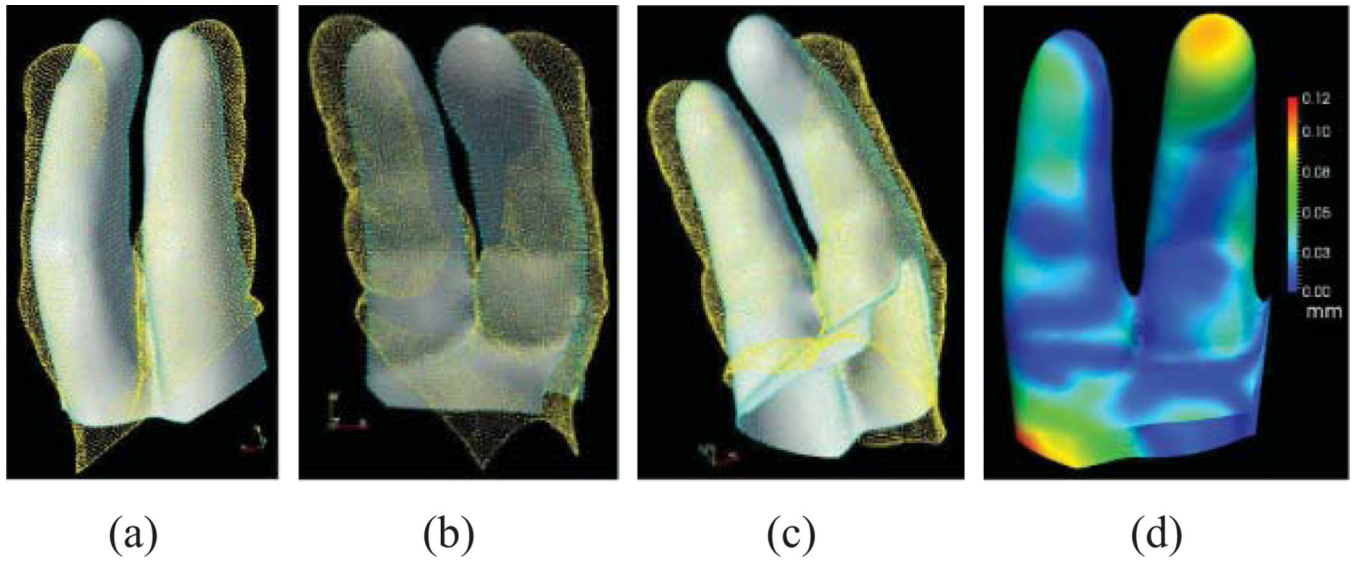


**Fig. 6.**  
RegEs after affine-PF-ICP, with 300 particles, and with noise.

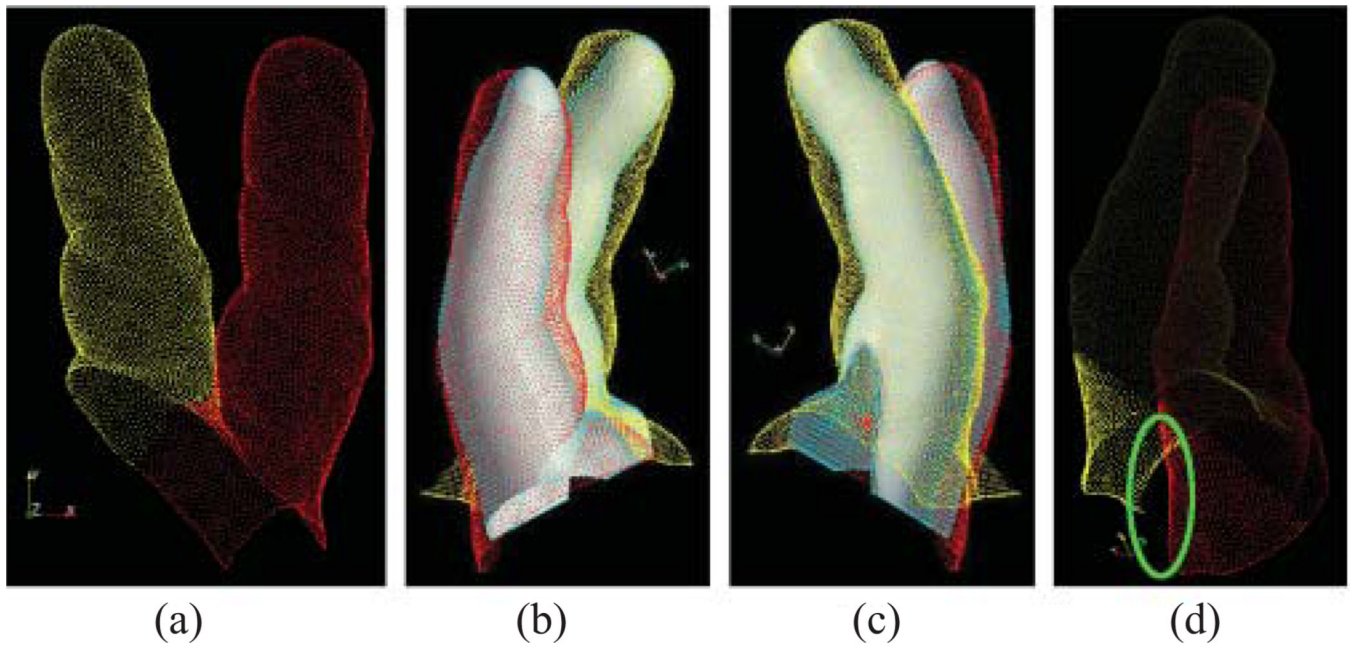


**Fig. 7.**  
(a)–(c) Two finger point sets in their original poses.



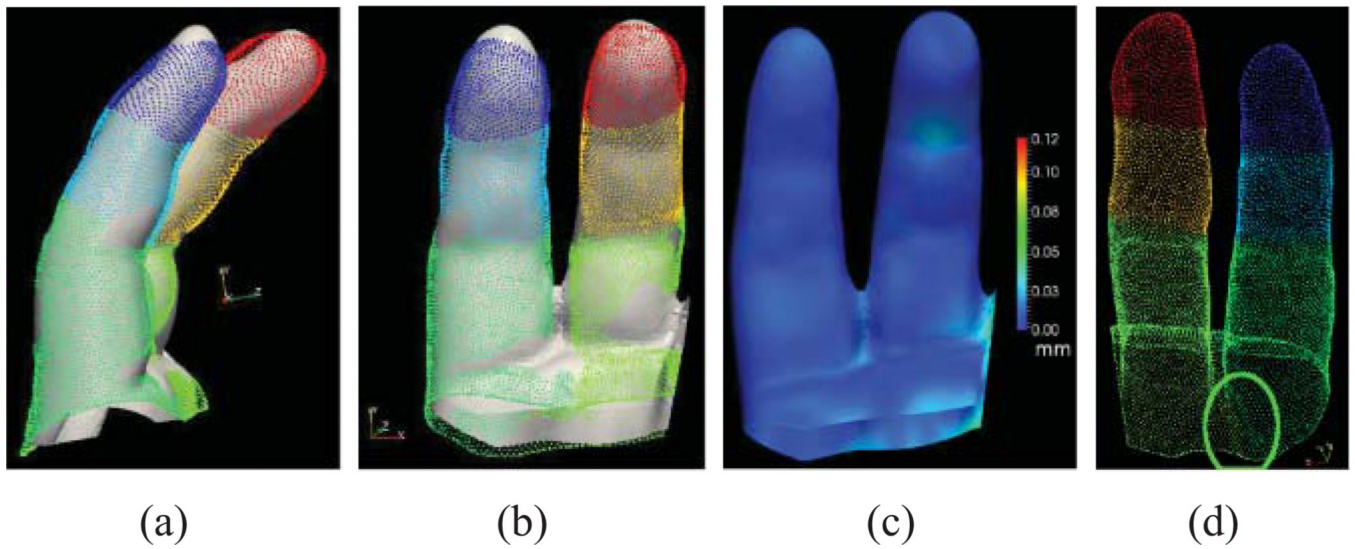
**Fig. 8.**

(a)–(d) Finger point set after affine-PF-ICP registration. Distance map to the closest point on the registered moving point set on the fixed point set. Warm color indicates large distance. Compare with Fig. 10(c), which uses the same color map scale.



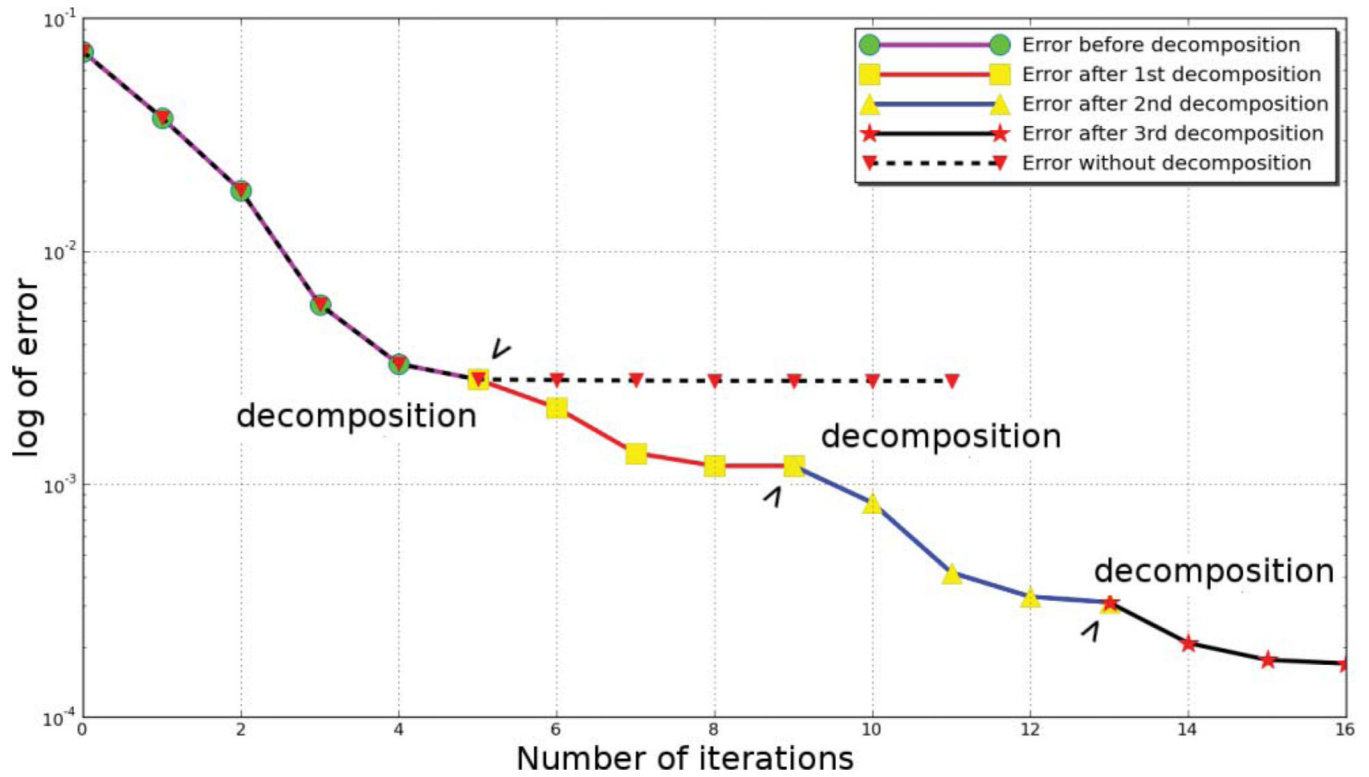
**Fig. 9.**

(a) Moving point set decomposed into two, with the two subsets registered to the fixed point set separately. (b) and (c) Compare subplot with Fig. 8(a) and (c) to see the angle difference being solved. (d) Subplot highlighting the tearing in region enclosed by the circle brought about by separate registration.

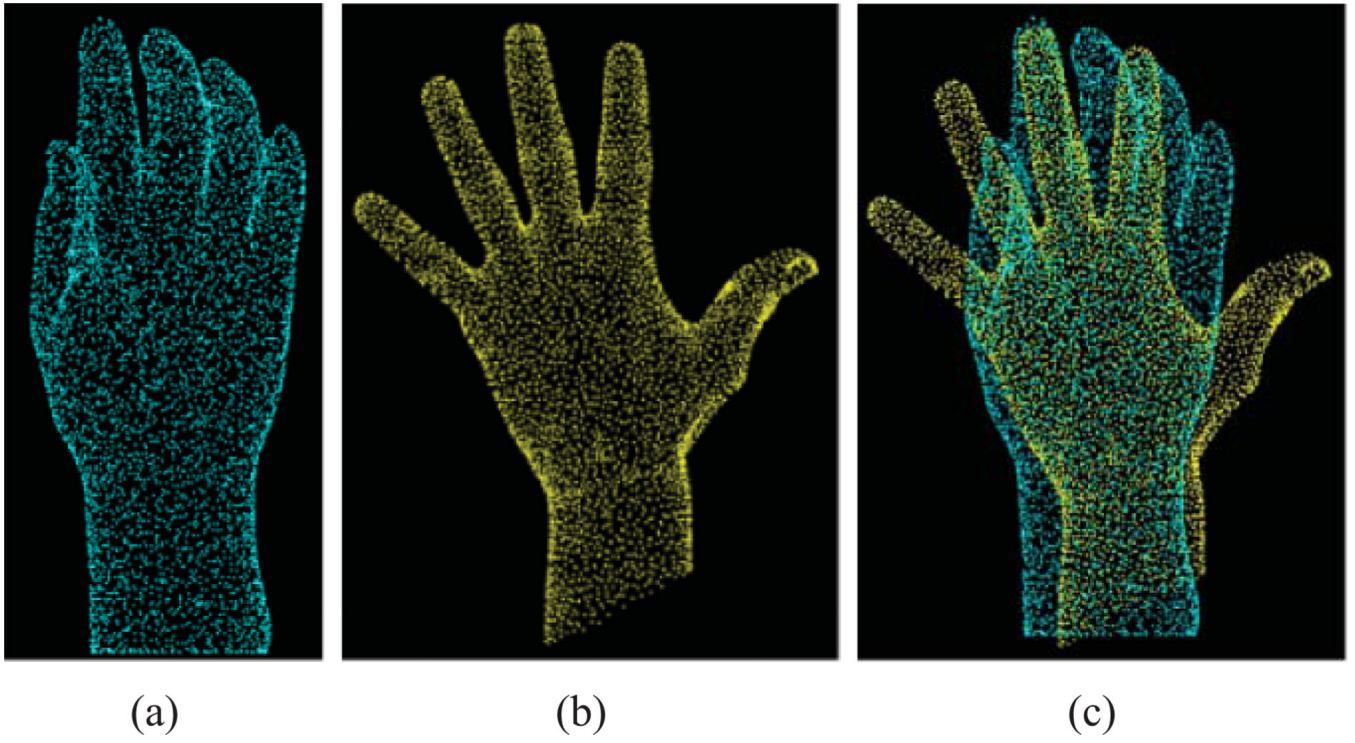


**Fig. 10.**

(a) and (b) Polyaffine registration result. Different colors indicate different groups formed by hierarchical decomposition. Compared to Fig. 9(b) and (c), the local deformations are well registered. (c) Distance map to the registered moving point set shown on the fixed point set. Compared to Fig. 8, the error is reduced. (d) Subplot highlighted in the circle shows “smooth stitching” compared to Fig. 9(d).

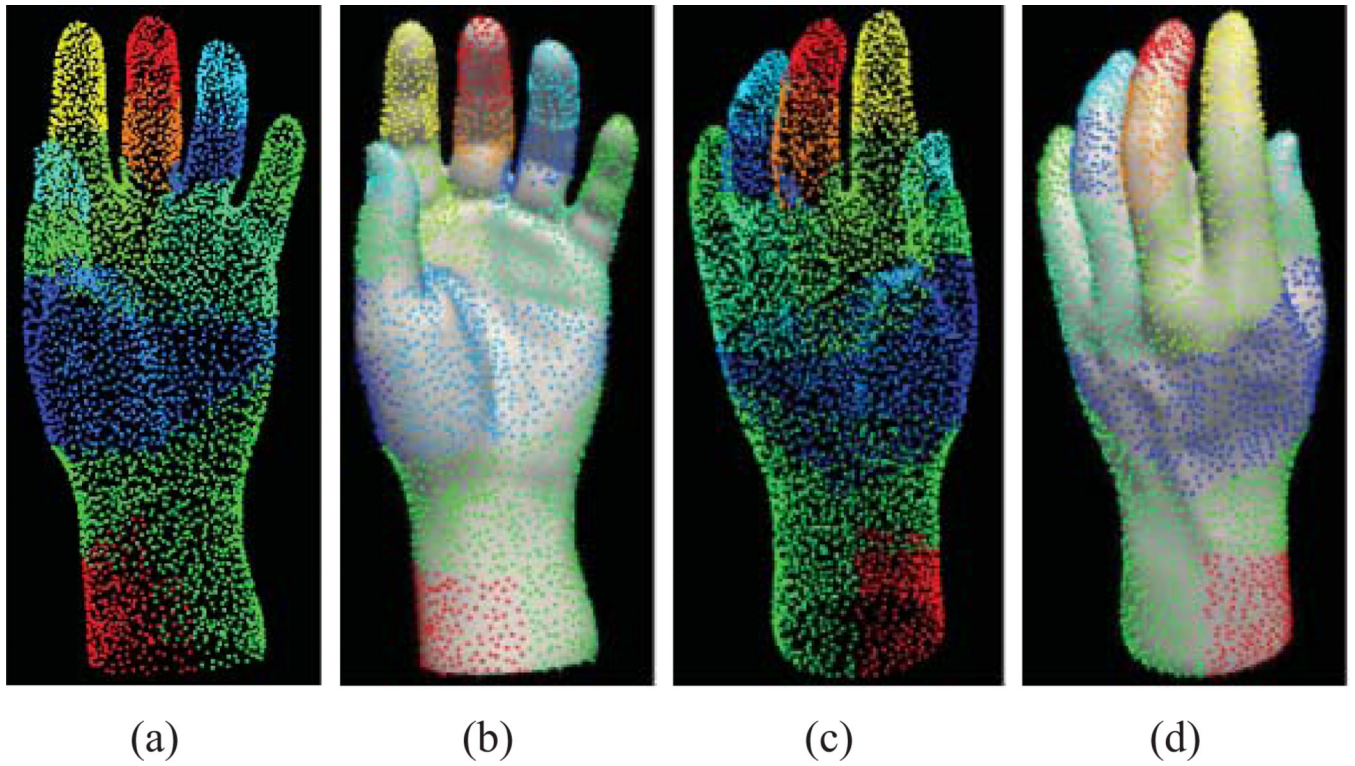
**Fig. 11.**

Error reduction through hierarchical decomposition-registration. Dashed line indicates error without decomposition (single affine-PF-ICP), that saturated after a certain number of iterations. The error is reduced with decomposition (indicated by arrows).



**Fig. 12.** Two hand point sets. (a) Fixed point set. (b) Moving point set. (c) Overlay fixed and moving point sets.

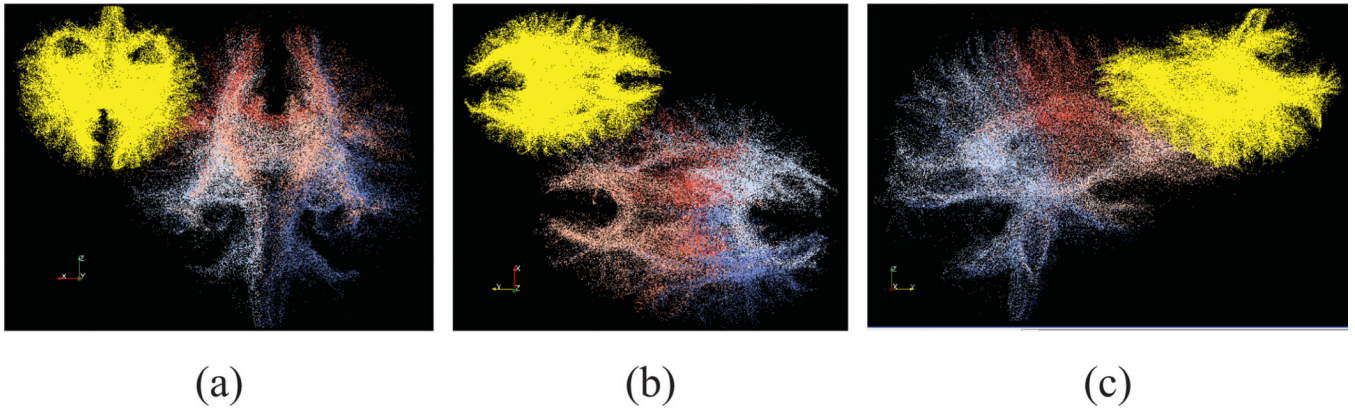




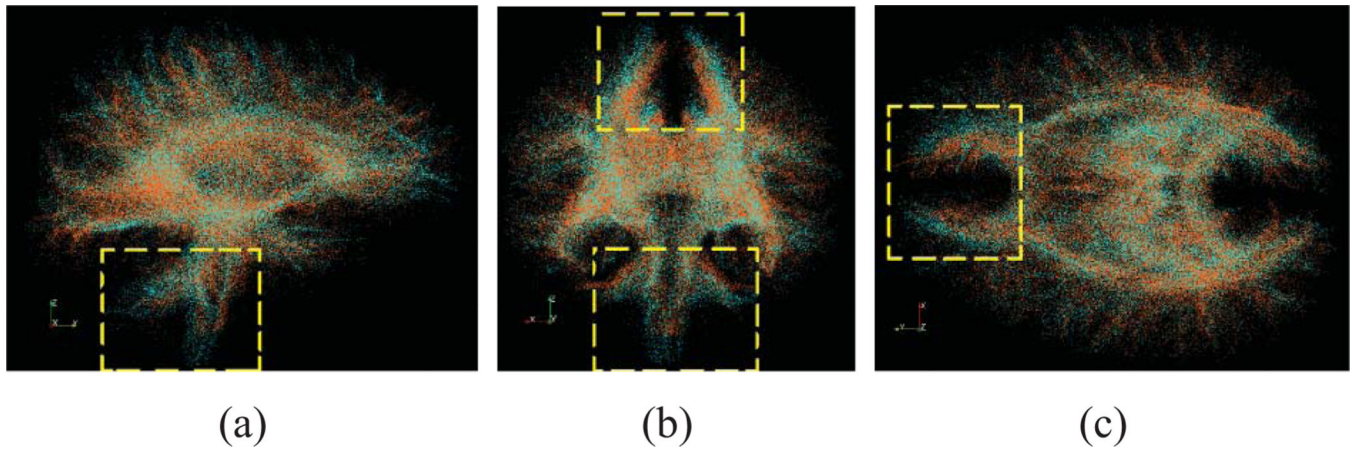
**Fig. 13.**

Registration result using the proposed method. (a) and (c) Registered moving point set, viewed from the front and back, respectively. (b) and (d) Views of (a) and (c) with the overlay of the fixed point set and its surface, for a clearer view.



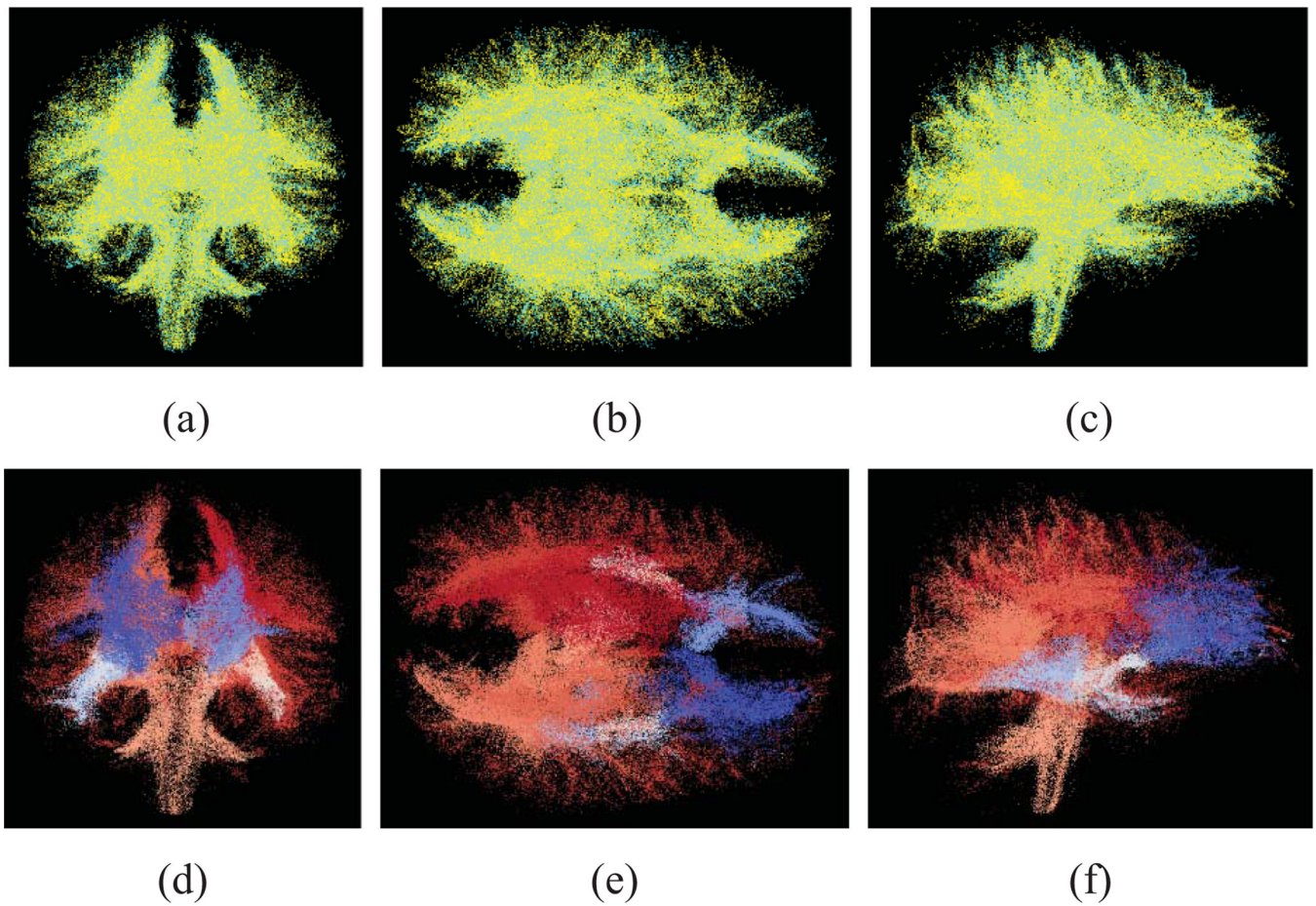
**Fig. 14.**

Clustered template point set (color) and unclustered moving point set (yellow), in their original pose and position. The fixed point set has the range:  $[-70.3, 66.7] \times [-87.0, 91.8] \times [-66.6, 73.0]$ . (a) Anterior view. (b) Superior view. (c) Right view.



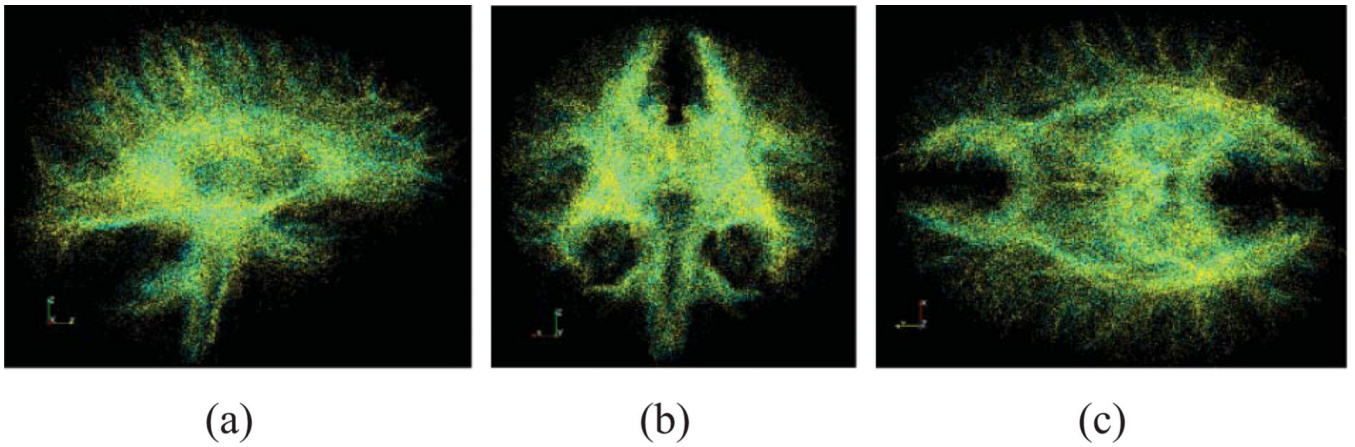
**Fig. 15.**

In all sub-plots, (a) sagittal, (b) coronal, and (c) axial, the affine-PF-ICP registered moving point set is in orange and the template point set is in cyan. See the mis-matching in the box.



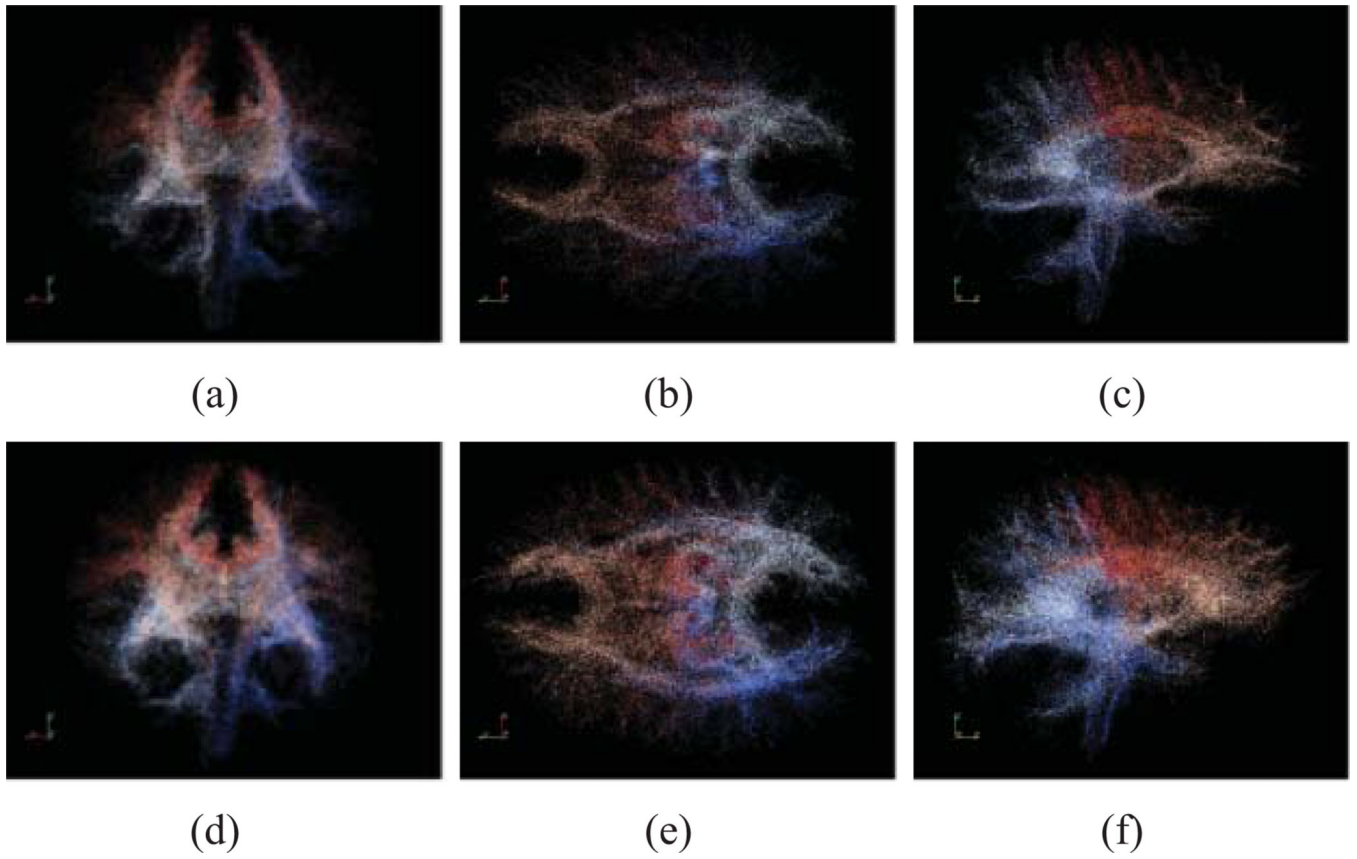
**Fig. 16.**

Top row: registration result of the proposed method on one data set. The fixed set is in cyan and the registered moving set is in yellow. Bottom row: clusters obtained in the process. Compare the regions corresponding to the boxes in Fig. 15 to see the local scale nonlinear deformation has been corrected. (a) Anterior view. (b) Superior view. (c) Right view. (d) Clustering anterior view. (e) Clustering superior view. (f) Clustering right view.



**Fig. 17.** Diffeomorphic registration using the clustering information of the fixed point set in (a) sagittal, (b) coronal, and (c) axial views. The fixed set is in cyan color and the registered moving set is in yellow. Compare the regions corresponding to the boxes in Fig. 15 to see the local scale nonlinear deformation has been corrected.





**Fig. 18.**

(a)–(c) Colors indicate clusters in template point set. (d)–(f) Moving point set became clustered while registered.

**TABLE I**

RegE of Registering the Hand Sets

Condition	Affine +non-rigid CPD	Proposed 30 particles	Proposed 100 particles	Proposed 300 particles
RegE	0.0082	0.05	0.0050	0.0032



**TABLE II**

Comparison of the Proposed Method When the Tracts are Manually Grouped

Set #	Set size	Final RegE		Running time	
		Proposed	Proposed, with clusters	Proposed	Proposed, with clusters
1	1,528,792	$8.90 \times 10^{-3}$	$7.75 \times 10^{-3}$	7.2 hr	30 min
2	1,176,057	$9.33 \times 10^{-3}$	$9.91 \times 10^{-3}$	6.5 hr	27 min
3	1,376,690	$8.81 \times 10^{-3}$	$7.10 \times 10^{-3}$	6.9 hr	31 min
4	1,614,770	$8.72 \times 10^{-3}$	$9.11 \times 10^{-3}$	7.5 hr	35 min