HHS Public Access

Author manuscript

IEEE Trans Pattern Anal Mach Intell. Author manuscript; available in PMC 2017 February 01.

Published in final edited form as:

IEEE Trans Pattern Anal Mach Intell. 2016 February ; 38(2): 238–251. doi:10.1109/TPAMI.2015.2448102

A Stochastic Approach to Diffeomorphic Point Set Registration With Landmark Constraints

Ivan Kolesov,
Department of Computer Science, Stony Brook University, Stony Brook, NY 11790

Jehoon Lee, Member IEEE,
Samsung Electronics Co., Ltd., Suwon, South Korea

Gregory Sharp, Member IEEE,
Department of Radiation Oncology at Massachusetts General Hospital, Boston, MA 02114

Patricio Vela, Member IEEE, and
Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332

Allen Tannenbaum, Fellow IEEE
Departments of Computer Science and Applied Mathematics, Stony Brook University, Stony Brook, NY 11790

Ivan Kolesov: ivan.kolesov@stonybrook.edu; Jehoon Lee: jehoon.lee25@gmail.com; Gregory Sharp: gcsharp@partners.org; Patricio Vela: pvela@gatech.edu; Allen Tannenbaum: arobertan@cs.stonybrook.edu

Abstract

This work presents a deformable point set registration algorithm that seeks an optimal set of radial basis functions to describe the registration. A novel, global optimization approach is introduced composed of simulated annealing with a particle filter based generator function to perform the registration. It is shown how constraints can be incorporated into this framework. A constraint on the deformation is enforced whose role is to ensure physically meaningful fields (i.e., invertible). Further, examples in which landmark constraints serve to guide the registration are shown. Results on 2D and 3D data demonstrate the algorithm's robustness to noise and missing information.

Index Terms

Point set; deformable registration; particle filter; simulated annealing; constrained optimization

I. Introduction

A problem that frequently arises in computer vision is the registration of two point sets. One seeks to obtain a transformation that maps a measured point set to a fixed model set. Point sets are often generated in the field of computer vision; for example, as output of a feature detector applied to an image [20], from higher level user input used to identify meaningful "landmarks" [12], [21], or as output of a scanning device. Once salient point features are extracted, registration on these point sets can be used for a variety of computer vision applications: determining stereo correspondences, image set matching [6] for panoramic stitching, surface registration [27], or medical imaging [43], [5]. Medical imaging benefits

from modeling an anatomical structure with a point cloud: the registration task is simplified and the computed deformation can be applied for atlas based segmentation.

Registration is an ill-posed problem and assumptions about the deformation field must be made or knowledge about the expected deformation must be available to compute a particular registration field. The non-rigid registration problem considered in this paper is challenging because the correspondences and the deformation are unknown; if the correspondences were known, one could perform landmark-based registration by fitting a smooth function to the data [38]. In [13], the authors propose using novel radial basis functions (RBFs), the Wendland functions, to perform landmark based registration; these functions have compact support and can capture fine deformations without affecting distant areas of the domain. Another sophisticated representation for the deformation field is the GRID model [16], [33].

Frequently, point sets undergoing registration are sparse representations of real objects. Physically, these objects cannot experience a deformation that is self-intersecting; thus, the field must be constrained to be diffeomorphic. Furthermore, prior information may be available (e.g., user input, rigidity constraints [29], etc.) and can provide strong clues to guide the registration algorithm. The proposed framework can easily incorporate these types of constraints. We focus on constrained, deformable registration and devise an algorithm that avoids local minima and limits the model complexity.

This paper is organized as follows. Section II reviews related works. Section III provides the background necessary for Section IV. The stochastic point set registration algorithm (SPSR) is presented in Section IV, with implementation details provided in Section V. The performance of the algorithm on 2D and 3D data sets is shown in Section VI. Section VII summarizes the results and describes future research directions.

II. Existing Work

Heuristic Methods

One of the best known and widely used point set registration algorithms is the iterative closest point (ICP) algorithm [3]. ICP is a popular rigid registration method for performing least squares minimization on the distance between two point sets; as for any gradient descent algorithm, a close initialization of the transformation must be provided to reach a global minimum. In fact, Besl and McKay [3] show that this iterative, heuristic approach has convergence properties similar to methods using explicit vector gradients, such as steepest descent. In [10], the authors propose a non-rigid robust point matching algorithm. The deformation is parameterized with thin plate splines (TPS) [19] and instead of assigning binary correspondences of data to model points, as ICP does, the authors use soft-assign [36]. Nevertheless, the iterative minimization approach in [10] is quite similar to ICP, but by using fuzzy correspondences and controlling the degree of the fuzziness with deterministic annealing, the authors obtain a more robust performance especially in the presence of outliers.

Gradient Descent Methods

Influence of outliers on the chosen energy functional is a concern for registration; in [40], Tsin and Kanade propose a cost function based on "kernel correlation," analyze its robustness to outliers, and demonstrate results with rigid registration examples. In the same spirit, Jian and Vemuri proposed the Robust Point Set (RPS) registration [24] that forms Gaussian mixtures from the model and data points and computes the integral of the squared difference between the mixtures. In [14], Glaunes *et. al* employed the large deformation framework to perform diffeomorphic registration on distributions and point sets. This variational framework represented the point sets as a summation of Dirac functions and used a kernel based norm to compute the distance between point sets. Wasserman *et. al* [44] proposed a novel definition of density for a point cloud: a mixture of non-uniform kernels whose shape is application specific. These densities are aligned with the optimization approach used in the state of the art *symmetric image normalization* (SyN) algorithm [2] for diffeomorphic image registration.

Probabilistic Methods

In [9], Chui and Rangarajan obtain the registration field by representing a template point set as a Gaussian mixture model and transforming the data points such that they are maximally explained by the template points. They extend the approach to the non-rigid case and perform optimization with the expectation maximization(EM) method. An elegant approach to unlabeled, diffeomorphic point set registration is proposed in [18], [17]. The authors use the deterministic EM approach to iteratively cluster points and estimate the diffeomorphism linking the cluster pairs. This approach gives excellent results if the point clouds have a high signal to noise ratio but it is expected that outliers will degrade performance because their influence is not accounted for explicitly. Most recently, Myronenko et al. [32] introduced the coherent point drift (CPD) algorithm. They replaced the use of TPS in [9] and parameterized the deformation with Gaussian functions instead. The algorithm can be used for data in dimension higher than three where TPS are not defined. Also, [32] removed the simulated annealing step controlling the search range around each model point within the EM algorithm by estimating the parameter instead.

Filtering Methods

The filtering approach to registration was introduced in [30] with the unscented particle filter (UPF) applied to rigid registration. The authors use the ICP algorithm to establish correspondences, compute the distance between data sets, and determine the likelihood for the UPF. The Euclidean distance underlying ICP is susceptible to outliers and hence, not always a reliable measure of fit between data sets. In [39], Sandhu *et al.* utilize particle filtering but select an inner product as a similarity measure between the two Gaussian mixtures (one is from the model point set and the other from the data point set). Outliers have local influence leading to a likelihood measure that is more robust. The authors also use a dynamical model to improve convergence speed and robustness. Their method only works for rigid registration. If elastic deformations are allowed, this similarity metric incorrectly moves the data set points toward the mode of the model set's Gaussian mixture density.

Contribution

In this work, a deformable registration framework is described. The approach presented here is not limited to a particular metric or parameterization for the deformation field. For evaluation, an energy for point set registration is selected, the one used in [24]. However, another cost function can replace it within the same optimization setting (e.g., the Fisher-Rao metric [35]); in fact, even differentiability of the function is not required. The only condition imposed is that the energy can be evaluated at any point in the domain.

Our work makes three contributions. First, we propose a stochastic approach based on simulated annealing for optimizing the cost function and introduce a generating function for the next state relying on particle filters; the merits of this optimization approach can be seen in Section VI-A where, using the same distance metric as RPS, SPSR achieves more accurate registration results because it find a better minimum for the energy. In the RPS and CPD approaches, the optimization is done by gradient descent and EM, respectively; both of these optimization techniques are known to be susceptible to local minima and the registration algorithms inherit their convergence properties.

Second, we demonstrate how the proposed optimization framework allows for the inclusion of a variety of constraints on the registration. Two types of constraints are presented, but other prior information can be easily incorporated in a similar fashion. First, the optimal deformation is restricted to be a diffeomorphism onto its range. Without the constraint, a deformation may not be feasible due to regions with overlap, see Fig. 2; hence, it is crucial in any application aiming to register physically meaningful data and makes the registration approach applicable to a wider class of problems. In other scenarios, it may be known that certain image regions can only deform rigidly while others undergo non-rigid deformations or that particular image areas should remain stationary. We show how to take into account this prior knowledge.

Third, we introduce a novel method to regularize the deformation field; instead of a parameter controlling global smoothness as in the RPS and CPD approaches, regularization is implicit through control of the number of basis functions. This difference is vital for accurate registration of data experiencing spatially varying deformations. This problem arises frequently in medical imaging where changes in anatomy are local, requiring a non-smooth deformation field while the remaining domain undergoes smooth, global changes where regularization is needed to avoid over-fitting. Gaussian radial basis functions (GRBFs) are frequently used to represent a deformation [42]; typically, the bases are uniformly distributed throughout the domain or located at all landmark points. In this formulation, the GRBFs composing the deformation are not restricted to being centered on data points, and the bases have independent covariance matrices, as opposed to [24] and [42]. This formulation allows a small number of basis function to be used in the representation.

III. Preliminaries

A. Point Set Registration Problem

Given a point set and a target point set, registration is the task of finding a mapping between the two sets with the appropriate properties. The target point set is called the model M with $M = \{m_1, ..., m_K\}$, and the given point set is called the data D with $D = \{d_1, ..., d_L\}$. The points $m_i, d_j \in \mathbb{R}^l$ where $l = 2$ or 3 for our experiments. The objective is to find a mapping $\mathscr{L}: \mathbb{R}^l \to \mathbb{R}^l$ that minimizes the distance, d, between the sets M and D:

$$\min_{\mathscr{L}} d(M, \mathscr{L}(D)). \quad (1)$$

B. Defining a distance metric

The distance d will depend on the representation chosen for the point sets. Here, the data and model sets are represented as kernel density estimates (KDEs), as in [23]. Identical, symmetric Gaussian kernels are placed around each point in the set to define the Gaussian mixture model for each point set,

$$m(\vec{x}; M) = \frac{1}{K} \sum_{i=1}^{K} \mathcal{N}(\vec{x} | m_i, \sigma^2 I), \quad (2)$$

$$c(\vec{x}; \mathscr{L}(D)) = \frac{1}{L} \sum_{j=1}^{L} \mathcal{N}(\vec{x} | \mathscr{L}(d_j), \sigma^2 I). \quad (3)$$

The point sets are maximally aligned when their corresponding kernel density estimates are maximally similar. A variety of information theoretic metrics exist to quantify the similarity between densities [24] and the SPSR framework presented in this work is not dependent on a particular choice. For the experiments, however, the L_2 or integrated square error (ISE) is used,

$$d(M, \mathscr{L}(D)) = \int_\Omega (m(\vec{x}; M) - c(\vec{x}; \mathscr{L}(D)))^2 d\Omega, \quad (4)$$

because a closed form expression for the distance can be computed by setting $\Sigma_T = \Sigma_1 + \Sigma_2$ from the identity:

$$\int_\Omega \mathcal{N}(\vec{x} | \vec{\mu}_1, \sum\nolimits_1) \mathcal{N}(\vec{x} | \vec{\mu}_2, \sum\nolimits_2) d\Omega = \mathcal{N}(\vec{\mu}_1 | \vec{\mu}_2, \sum\nolimits_T). \quad (5)$$

C. Parameterizing the Displacement Function

The space of non-rigid transformations is infinite-dimensional, which complicates the optimization of Eq. (1). To improve the tractability of the optimization, we limit the degrees of freedom of the deformation field $\mathscr{L}$ by defining the transformation to be an additive

composition of a rigid transformation and a collection of non-rigid basis transformations through a GRBF network,

$$\mathscr{L}(\vec{x};\Theta)=\underbrace{A\vec{x}+\vec{b}}_{\mathscr{L}_{lin}}+\underbrace{\sum_{i=1}^{N}\vec{w}_i\mathscr{N}(\vec{x}|\vec{\mu}_i,\sigma_i^2 I)}_{\mathscr{L}_{nl}}. \quad (6)$$

The optimization in Eq. (1) is now over the rigid transformation parameters and the GRBF parameters. Finding the optimal $\mathscr{L}$ is equivalent to determining the parameters Θ:

$$\begin{gathered}\Theta=[\Theta_r,\Theta_g] \quad \text{where}\\ \Theta_r=\left[A_p,\ \vec{b}\right]\\ \Theta_g=[\vec{\mu}_1,\vec{w}_1,\sigma_1,\cdots,\vec{\mu}_N,\vec{w}_N,\sigma_N] \quad (7)\\ A_p=[w,x,y,z] \quad \vec{b}=[a,b,c]\\ \vec{\mu}_i=[\mu_{x_i},\mu_{y_i},\mu_{z_i}] \quad \vec{w}_i=[w_{x_i},w_{y_i},w_{z_i}]\end{gathered}$$

for the 3D case (the 2D case is simpler). The parameters A_p are quaternion coordinates that generate the scaled, rigid transformation matrix A. By representing $\mathscr{L}(\cdot)$ as in Eq. (6), the problem in Eq. (1) becomes one of parameter estimation:

$$\min_{\Theta} E(\Theta) \text{ with } E(\Theta)=d(M,\mathscr{L}(D;\Theta)). \quad (8)$$

The representational capacity of the deformation by a GRBF network is guaranteed by the *universal approximation theorem* for radial basis function networks.

Theorem: [34] Let the family $S_{\mathscr{K}}$ consist of functions $q:\mathscr{R}^r\to\mathscr{R}$,

$q(x)=\sum_{i=1}^{N} w_i\cdot\mathscr{K}(\frac{x-t_i}{\sigma})$, $N\in\mathscr{N}$, $\sigma>0$, $w_i\in\mathscr{R}$, $t_i\in\mathscr{R}^r$. Let $\mathscr{K}:\mathscr{R}^r\to\mathscr{R}$ be an integrable bounded function such that $\mathscr{K}$ is continuous almost everywhere and $\int_{\mathscr{R}^r}\mathscr{K}(x)dx\neq 0$. Then family $S_{\mathscr{K}}$ is dense in $L^p(\mathscr{R}^r)$ for every $p\in[0,\infty)$.

The symmetric Gaussian function satisfies the properties required of $\mathscr{K}$. Relaxing the constraint of equal σ, it is still possible to find an N such that $\|f-\mathscr{L}\|_{L^2}<\varepsilon$, for a given ε. The number of basis functions N in the term $\mathscr{L}_{nl}$ of Eq. (6) regularizes the deformation. Since registration is an ill-posed problem, changing the regularization (varying N) leads to different deformation fields. Although with a sufficiently high N the linear term in Eq. (6) is unnecessary, it is included to reduce the number of GRBFs needed; the rigid transformation terms account for global movement and the GRBF network represents local deformations. While a large N would lead to increasingly accurate non-rigid registration, we use a small to moderate value of N (10 to 20) to limit variation in the transformation and prevent fitting to noise or other sources of error.

D. Particle Filtering

Sequential Bayesian filtering estimation with Monte Carlo simulation, called *particle filtering*, was first introduced by Gordon [15]. In recent years, it has proven to be a powerful

scheme for non-linear and non-Gaussian estimation problems due to its simplicity and versatility.

It is common to receive measurements at discrete time points k, and the discrete-time filtering problem is formulated as follows. Let x_k be an unobservable state and $x_{1:k}$ be the history of states up to time k; similarly, let y_k be observable measurements and $y_{1:k}$ be the history of these measurements. Then, the transition equation Eq. (9) and the measurement equation Eq. (10) for the general Markov state-space model are

$$x_k = f_k(x_{k-1}, u_{k-1}), \quad (9)$$

$$y_k = h_k(x_k, v_k) \quad (10)$$

and the conditional distributions are

$$X_k | (X_{k-1} = x_{k-1}) \sim p(x_k | x_{k-1}) \quad (11)$$

$$Y_k | (X_k = x_k) \sim p(y_k | x_k). \quad (12)$$

Here, f_k, h_k are (potentially) time-varying non-linear functions and u_k, v_k are independent and identically distributed (iid) random variables representing noise in the state and measurement equations, respectively, with known probability density functions. Thus, Eq. (9) and Eq. (10) implicitly define the state transition and measurement probabilities $p(x_k | x_{k-1})$ and $p(y_k | x_k)$, respectively.

In applying the PF for registration, the history of states is not important and the aim is to estimate $p(x_k | y_{1:k})$, the marginalized version of $p(x_{1:k} | y_{1:k})$. To begin, N samples are drawn from initial state distribution, $p(x_0)$ and a transitional prior is assumed. Then, the algorithm can be decomposed into two steps [8]:

Predict: Chapman-Kolmogorov Equation

$$p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) \quad (13)$$

Update: Bayes Theorem

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})}. \quad (14)$$

Thus, the equation incorporating a new measurement y_k is

$$p(x_k | y_{1:k}) \propto p(y_k | x_k) \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}).$$

E. Particle Filtering for Global Optimization

The static problem of finding the global minimum $x^*=\operatorname{argmin}_x R(x)$ through stochastic optimization can be phrased in a principled manner as a Bayesian filtering problem discussed in Section III-D. One formulation is presented in [45]; below, we present a different formulation. To this end, the following state-space model is defined:

$$x_{k+1}=d(x_k)+u_k \quad (15)$$

$$y_k=R(x_k)+v_k \quad (16)$$

where u_k, v_k are i.i.d. random variables $u_k \sim \mathcal{N}(0,Q)$ and $v_k \sim \mathcal{N}(0, 1)$. These variables represent the uncertainty in the state update and the state measurement; they implicitly define the probabilities

$$p(x_k|x_{k-1})=\frac{1}{\left((2\pi)^d|Q|\right)^{\frac{1}{2}}}e^{-\frac{1}{2}\|(x_k-d(x_{k-1}))\|^2_{Q-1}} \quad (17)$$

$$p(y_k|x_k)=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}(y_k-R(x_k))^2}. \quad (18)$$

It is assumed that the measurements y_k are equal to the optimal value of $R(x)$, i.e., $y_k = R(x^*)$ $\forall k \in \{0, 1, ...\}$. Of course, x^* is unknown *a priori* as is $R(x^*)$. Instead, $y_k = g$ is used where g is a lower bound for $R(x)$:

$$g \leq R(x),\ \forall x \in S \text{ where } S=\text{domain of } R(x). \quad (19)$$

From Eq. (18), if $R(x_k^i)<R(x_k^j)$ then $p(y_k|x_k^i)>p(y_k|x_k^j)$. Thus, as $k\to\infty$, the filter converges to

$$p(x_k|y_{1:k})=\delta(x_k-x^*). \quad (20)$$

In the case of Eq. (8), an obvious choice for g is $g = 0$ since the integral L_2 distance is always positive. A trivial choice for the state dynamics, $d(\cdot)$, is $d(x_k) = x_k$, which corresponds to a random walker, according to Eq. (15). More sophisticated local exploration approaches can be used (e.g., [22]) to explore the state space efficiently.

Stochastic optimization is naturally formulated as state estimation of a dynamical system. The particle filtering framework allows for arbitrary system dynamics, given a state x_k it is sufficient to know the resulting x_{k+1}, and the only requirement on the measurement function $h_k(\cdot)$ is that it can be evaluated. This flexibility makes inclusion of constraints straightforward, with little modification to the optimization framework, and allows one to easily incorporate prior knowledge, such as severity of the deformation, by modifying the

system dynamics. On the other hand, this field is mature and there are strong convergence results guaranteeing that the sequential Monte Carlo method will solve the filtering problem.

F. Simulated Annealing

Simulated annealing is a Markov Chain Monte Carlo (MCMC) method for global optimization [26] that samples the sequence of probability distributions $\{p_k(x)\}_{k\in\mathbb{T}}$ with $\mathbb{T}=\{1, ..., b\}$,

$$p_k(x) \propto [p(x)]^{\frac{1}{T_k}} \quad \text{where } 0<T_b<\ldots<T_1 \text{ and } 1 \ll \tfrac{1}{T_b}. \tag{21}$$

Suppose it is desired to obtain the global maximum of a distribution $p(x)$. One inefficient solution is to sample from $p(x)$ using a Markov chain Monte Carlo (MCMC) approach to produce samples $X^i \sim p(x)$ for $i \in [1, ...,N]$ and choose the maximum according to

$$X^*=\underset{X_i}{\operatorname{argmax}}\, p(X^i). \tag{22}$$

Unless $p(x)$ has significant probability mass near the optimum, the sampler will likely spend much of the computation time in regions distant from the global mode. Instead, an annealing approach can be used: samples are taken from the sequence of pdfs in Eq. (21) to encourage sampling from around the global maxima of $p(x)$ [11] as $T_b \to 0$.

Simulated annealing (SA) is the basis of the stochastic method proposed in this work to find the minimum x^* of an objective function $R(x)$. To this end, we set $p(x) \propto e^{-R(x)}$ in Eq. (21); then, as $b\to\infty$ samples are drawn from around the global maxima of $p(x)$ [11]. Extensions to continuous variables are described in [28], [41]. The general continuous simulated annealing (CSA) algorithm for finding a globally optimal solution is presented in Algorithm 1.

Algorithm 1

Continuous Simulated Annealing.

1: Initialize state variable: Θ_0, $s_0 = \{\Theta_0\}$, $k = 0$, $t_0 = 1$

2: Generate the candidate state: $\tilde{\Theta}_{k+1} \sim G(\Theta_k,\Theta_{k+1})$

3: Accept/reject candidate state with the following rule:

$$\theta_k = \begin{cases} \tilde{\Theta}_{k+1} & p \le \mathscr{A}(\Theta_k, \tilde{\Theta}_{k+1}, t_k) \text{ (i.e. accepted)} \\ \Theta_k & \text{otherwise} \quad \text{(i.e. rejected)} \end{cases}$$

Generate p from a uniform distribution $\mathscr{U}(0, 1)$

t_k is temperature of annealing at time k

4: Record traversed states until time k: $s_{k+1} = s_k \cup \{\Theta_k\}$

5: Cooling schedule determines t_k, $t_{k+1} = U(k + 1)$

6: If stopping criteria not reached, $k = k + 1$ go to Step 1

SA is a Metropolis-Hastings (M-H) algorithm for which the stationary distribution $p_k(x)$ is defined in Eq. (21). The acceptance function is defined as

$$\mathscr{A}(x_k, \tilde{x}_{k+1}, T_k) = \min\left\{1, \frac{p^{\frac{1}{T_i}}(\tilde{x}_{k+1})G(x_k|\tilde{x}_{k+1})}{p^{\frac{1}{T_i}}(x_k)G(\tilde{x}_{k+1}|x_k)}\right\},$$

and if the transition kernel is reversible (i.e., $G(x_k|\tilde{x_{k+1}}) = G(\tilde{x_{k+1}}|x_k)$), it can be simplified to

$$\mathscr{A}(x_k, \tilde{x}_{k+1}, T_k) = \min\left\{1, \frac{p^{\frac{1}{T_i}}(\tilde{x}_{k+1})}{p^{\frac{1}{T_i}}(x_k)}\right\}.$$

In the M-H formulation [37], [1], the detailed balance, which leads to an acceptance function requiring a reversible kernel, is stated as follows

$$p(x_k)G(x_{k-1}|x_k) = p(x_{k-1})G(x_k|x_{k-1}).$$

This condition is sufficient (not necessary) to ensure that the function from which the MCMC is sampling remains the desired invariant distribution $p(x)$ since

$$p(x_k) = \int p(x_{k-1})G(x_k|x_{k-1})dx_{k-1}. \quad (23)$$

However, since the temperature T_k is being changed at each time, Eq. (23) does not hold and maintaining detailed balance by restricting the transition kernel to be reversible is not necessary. We relax this constraint.

Furthermore, simulated annealing can be seen as sampling from a sequence of distributions $p_k(x) = p^{\frac{1}{T_k}}(x)$ [11] using a single particle and the SMC framework. There are no restrictions placed on the transition kernel. And, since sampling from the sequence $p_k(x)$ exactly is not necessary for finding the optimum, the only desired property is that the chosen kernel asymptotically samples from around global maxima of $p(x)$.

IV. Constrained, Stochastic Point Set Registration Algorithm

A. Examples of Useful Constraints

A measure of similarity between two point sets was presented in Section III-A; along with other examples in [24], [10], [32], the corresponding algorithms optimize the unconstrained problems to achieve registration. Registration is an ill-posed problem and the commonly used Tikhonov approach is not the only way or the best way to regularize the problem. This work focuses on meaningful regularization of the deformation field by adding constraints to the existing objective functions (e.g., Eq. (8)). Constraints are useful for enforcing smoothness(e.g., maintaining a positive definite Jacobian) and restricting the allowed deformations. We recognize these needs and the optimization approach proposed in Section IV-B is well suited to include constraints. In this section, two examples of frequently

applicable constraints are presented; they are not required to execute SPSR but are often useful in the types of problems that are considered. Also, the constraints are not the only ones possible within the proposed registration framework but are meant to serve as examples.

The first type of constraint is the injectivity constraint. The data points considered in this work, are sparse representations of real objects with physical meaning (e.g., landmarks in a medical image). Hence, the deformation should align the two sets as well as possible while respecting physical constraints. One such constraint is that two different points in space cannot map to the same point. When the constraint fails, the deformation field appears to overlap itself, as per Fig. 2. This constraint requires a one-to-one mapping of the domain under $\mathscr{L}$. A second constraint is that the transformation must preserve the orientation of the space. These constraints are not strictly enforced by any state-of-the-art registration algorithms discussed in Section II. Uniqueness of the mapping and preservation of orientation is tested through the Jacobian J of $\mathscr{L}$ from Eq. (6). In particular, enforcing the constraint

$$d_J(\vec{x}) = \det(J(\mathscr{L}(\vec{x};\Theta))) > 0 \quad , \forall \vec{x} \in S, \quad (24)$$

will ensure an orientation preserving and one-to-one transformation [31]. The function det(·) is the determinant, and $S \subset \mathrm{R}^l$ is an open subset containing the region of interest.

Another constraint arises from the necessity to keep certain points/regions in the domain stationary while the remainder are free to move in the direction minimizing the energy functional. The local rigidity constraint [29] states that points in a subdomain can undergo only a rigid deformation and points outside can move non-rigidly. This constraint can be reduced to the problem of fixing points in a subdomain to be stationary by: rigidly aligning the point sets according to the constrained regions and performing deformable registration subject to the deformation being the identity within these regions. The next example of constraints that can be incorporated into the proposed optimization framework is a landmark constraint. Suppose two subsets of corresponding points, each containing C points, have been identified: $M_c = \{m_1, ..., m_C\}$ from the model set and $D_c = \{d_1, ..., d_C\}$ from the data set. First, we register the full point sets using just the landmark correspondences (e.g., [5]) to find the deformation $\mathscr{L}_c(x)$; applying this deformation to the moving set D_c, it is guaranteed that $\mathscr{L}_{co}(d_i) = m_i$ for $i \in [1, ...,C]$. After landmark based registration, the resulting model set M and data set D will be aligned using the proposed approach subject to the constraints that the aligned landmark points $D_c = \{d_1, ..., d_C\} = \{\mathscr{L}_{co}(d_1), ...,\mathscr{L}_{co}(d_C)\} = M_c$ do not move. Explicitly, user constraints that prevent chosen points $\{d_1, ..., d_C\}$ from moving are stated as $\mathscr{L}(d_i;\Theta) = d_i$ for $i \in [1, ...,C]$. The optimization problem becomes

$$\min_{\Theta} E(\mathscr{L}(\vec{x};\Theta)) \quad (25)$$

$$s.t. \quad \begin{aligned} \mathscr{L}(\tilde{d}_1;\Theta)&=\tilde{d}_1 \\ &\vdots \\ \mathscr{L}(\tilde{d}_C;\Theta)&=\tilde{d}_C \end{aligned} \tag{26}$$

$$\det(J(\mathscr{L}(\vec{x};\Theta))>0. \tag{27}$$

We add Gaussian basis functions of width σ_C^2 centered at the points $\{\tilde{d_1}, ..., \tilde{d_C}\}$ to the original parameterization of $\mathscr{L}_{nl}$ in Eq. (6):

$$\hat{\mathscr{L}}(\vec{x};\Theta)=A\vec{x}+b+\mathscr{L}_{nl}(\vec{x};\Theta)+\mathscr{L}_c(\vec{x}) \text{ where} \tag{28}$$

$$\mathscr{L}_c(\vec{x})=\sum_{i=1}^{C}\vec{w}_{C_i}\mathscr{N}\left(\vec{x}|\tilde{d}_i,\sigma_C^2 I\right) \tag{29}$$

and $\vec{w_{Ci}}$ satisfy

$$\begin{aligned} \sum_{i=1}^{C}\vec{w}_{C_i}\mathscr{N}\left(\tilde{d}_i|\tilde{d}_i,\sigma_C^2 I\right)&=-\mathscr{L}_{nl}(\tilde{d}_1;\Theta) \\ &\vdots \\ \sum_{i=1}^{C}\vec{w}_{C_i}\mathscr{N}\left(\tilde{d}_C|\tilde{d}_i,\sigma_C^2 I\right)&=-\mathscr{L}_{nl}(\tilde{d}_C;\Theta). \end{aligned} \tag{30}$$

The weights for these new basis functions $\vec{w_{Ci}}$ depend on the parameters Θ and are computed such that $\hat{\mathscr{L}}(\tilde{d_i};\Theta) = \tilde{d_i}$ for $i \in [1, ...,C]$. When $\hat{\mathscr{L}}$ is equal to the identity, $\vec{w_{Ci}} = 0\vec{}$. As $\hat{\mathscr{L}}$ changes from the identity at the constraint point, the linear system of equations in Eq. (30) is solved for $\vec{w_{Ci}}$.

To enforce some/all of the constraints from Eq. (26)–(27), optimization of a cost function $R(\Theta)$ is performed instead of the unconstrained similarity metric in Eq. (8). Three choices for $R(\Theta)$ are presented in Eq. (31)–(33)

$$R_1=E(\mathscr{L}(D;\Theta)) \tag{31}$$

$$R_2=\begin{cases} E(\mathscr{L}(D;\Theta)) & r(\cdot)>0 \\ B\cdot(\|m\|^2+\|c(\vec{x};\mathscr{L}(D;\Theta))\|^2) & r(\cdot)\le 0 \end{cases} \tag{32}$$

$$R_3=\begin{cases} E\left(\hat{\mathscr{L}}(D;\Theta)\right) & r(\cdot)>0 \\ B\cdot(\|m\|^2+\|c(\vec{x};\hat{\mathscr{L}}(D;\Theta))\|^2) & r(\cdot)\le 0 \end{cases} \tag{33}$$

where $r(\Theta) = \det(J(\mathscr{L}(x;\Theta)))$, m is the function $m(x;\vec{M})$ from Eq. (2), and the rationale for these choices is described in the subsequent paragraph.

First, for unconstrained optimization of Eq. (8), $R(\Theta) = R_1(\Theta)$ is chosen. Second, the injectivity constraint only, in Eq. (24), is enforced by selecting $R(\Theta) = R_2(\Theta)$ for optimization. In SPSR, optimization of the function $R(\Theta)$ is performed using a particle filter, as described in Section III-E. The cost function $R_2(\Theta)$, in Eq. (32), imposes a soft constraint on the injectivity of $\mathscr{L}$. This soft constraint holds exactly as $B \to \infty$: if $r(x_n) \leq 0$, then, $p(y_n|x_n) \approx 0$, which leads to the rejection of candidate particles resulting in non-physical deformation fields by setting their likelihood to zero. Finally, choosing $R(\Theta) = R_3(\Theta)$ ensures that the optimal deformation will be invertible and will satisfy the landmark constraints in Eq. (26) by evaluating the likelihood of Θ using the deformation field $\hat{\mathscr{L}}(x;\Theta)$.

B. Optimization Algorithm

A gradient descent method cannot be employed to minimize the cost functions in Eq. (32)–(33) because they are not differentiable and a satisfactory initialization is unavailable. Instead, a stochastic, hierarchical approach is used to find the parameters in Eq. (7). The similarity transformation parameters (rigid rotation, scale, and translation) Θ_s are computed first. Since the dimensionality is low, the global PF optimization from Section III-E is used directly to find

$$\hat{\Theta}_s = \min_{\Theta_s} R(\Theta_s|\Theta_g). \quad (34)$$

Now, we introduce the stochastic point set registration (SPSR) algorithm to perform the following optimization:

$$\hat{\Theta}_g = \min_{\Theta_g} R(\Theta_g|\hat{\Theta}_s). \quad (35)$$

The overall strategy is shown in Fig. 3, whose details follow. It is known that the sufficient number of particles for estimating distributions using a particle filter grows exponentially with the dimensionality of the state space. The parameter vector Θ_g of $R(\cdot)$ is large; there are seven parameters per basis function. However, while parameters belonging to the same basis are highly related, there is a weaker correlation between parameters of different bases. Blindly applying a particle filter to correlated variables does not take into account this prior information about their dependence. Hence, rather than optimizing over the entire state vector Θ_g in Eq. (7), we group elements of the state vector into zones as

$$\begin{aligned} \Theta_g &= \left[\vec{\theta}_1, \ldots, \vec{\theta}_N\right] \quad \text{where} \\ \vec{\theta}_i &= [\vec{\mu}_i, \vec{w}_i, \sigma_i] \quad \text{for } i \in [1, N]. \end{aligned} \quad (36)$$

and minimize $R(\cdot)$ with respect to a single zone $\vec{\theta}_i$ at a time:

$$\vec{\theta}^*_i = \underset{\vec{\theta}_i}{\operatorname{argmin}}\, R\left(\vec{\theta}_i|\Theta_{\neg i}, \hat{\Theta}_s\right). \quad (37)$$

The optimization in Eq. (37) is accomplished over the lower dimensional space with the PF, as per Section III-E. In this case, the state x_k being estimated is $x_k = \vec{\theta_i}$.

The optimum with respect to the entire parameter vector Θ_g is achieved using the CSA approach described in Section III-F; this section details the functions $\mathscr{A}(\cdot)$, $G(\cdot, \cdot)$, and $U(\cdot)$. The generator function $G(\tilde{\Theta}_{k+1},\Theta_k)$ determines how the transition from the current state Θ_k to a proposed state $\tilde{\Theta}_{k+1}$ is made. The particular generator function used for SPSR is defined by Algorithm 2. Here, generating a proposed state involves perturbing a single zone in Eq. (36) while keeping the others constant. In particular, at the k^{th} iteration, this perturbation is made by optimizing the objective function over parameters in zone $i = \mathrm{mod}(k,N)$, as written in Eq. (37), which correspond a single Gaussian basis function.

Algorithm 2

The generator function $G(\tilde{\Theta}_{k+1},\Theta_k)$.

1: Compute $i = \mathrm{mod}(k,N)$,

2: Set $\Theta_{\neg i} = [\vec{\theta_1}, ..., \vec{\theta_{i-1}}, \vec{\theta_{i+1}}, ..., \vec{\theta_N}]$

3: Use PF (Section III-E) to find $\vec{\theta}^*_i = \underset{\vec{\theta}_i}{\operatorname{argmin}} R\left(\vec{\theta}_i | \Theta_{\neg i}, \hat{\Theta}_s\right)$

4: Set $\tilde{\Theta}_{k+1} = \left[\vec{\theta}_1, \ldots, \vec{\theta}_{i-1}, \vec{\theta}^*_i, \vec{\theta}_{i+1}, \ldots, \vec{\theta}_N\right]$

The acceptance function used is the Metropolis function

$$\mathscr{A}(\Theta_k, \tilde{\Theta}_{k+1}, t_k) = \min\left\{1, e^{-\frac{R([\hat{\Theta}_s, \tilde{\Theta}_{k+1}]) - R([\hat{\Theta}_s, \Theta_k])}{t_k}}\right\},$$

, and the cooling schedule chosen is

$$U(k) = \frac{R(\left[\hat{\Theta}_s, \Theta_0\right])}{1 + (\lceil \frac{k}{N} \rceil)^2}$$

where the $\lceil \cdot \rceil$ is the ceiling function. The generator function proposed in this section is clearly not reversible, $G(\Theta_{k+1},\Theta_k) \neq G(\Theta_k,\Theta_{k+1})$. In Metropolis-Hastings (M-H), detailed balance is maintained by construction to ensure sampling from a time-invariant distribution. In simulated annealing, since homogeneity is violated due to the time dependent probability distribution, detailed balance is no longer maintained [1]. Thus, there is no requirement to enforce detailed balance through a symmetric transition kernel [25]. We cannot prove that the exact global minimum will be found with the chosen generator function and cooling schedule, but the proposed algorithm can accept states that cause the cost function $R(\cdot)$ in Eq. (8) to increase and thus, escape local minima. Also, results in Section VI-A empirically

show that better minima are found, since the algorithm outperforms other state of the art registration approaches.

C. Illustration of the Registration Process

This section demonstrates the effect of changing the number of basis functions N that are used to represent the deformation field in Eq. (6) and illustrates the registration process presented in Section IV-B. An example of two misaligned point sets of a giraffe are shown in Fig. 4(a); misalignment is large around the head of the animal and becomes increasingly small moving towards the tail. In this experiment, three values of N were used $N = 2, 6, 10$, and a registration using $R_2(\Theta)$ from Eq. (32) was run to convergence for each value of N. The results are shown in Fig. 4(b)–4(d), respectively. It is expected that misalignments making the largest contribution to the cost function will be corrected first, and increasing the degrees of freedom by raising N will allow the algorithm to capture small errors in alignment. Results of the experiment in Fig. 4 support this intuition: with just two basis functions in the deformation, the majority of misalignment is captured. Increasing the number of basis functions to six captures smaller contributions to the energy function by aligning the back and legs of the giraffe. Finally, with ten basis functions, the point set are almost perfectly aligned; small errors are present around the giraffe's ears, which, if desired, can be corrected by increasing N further. For most applications, the results in Fig. 4(d) would be deemed sufficiently close and increasing N unnecessary.

The sequential nature of the SPSR registration algorithm is illustrated in Fig. 5; this figure should be read simultaneously with Fig. 3. Each subfigure 5(a)–5(f) is the result of generating a state at time k, Θ_k and warping the domain according to the deformation $\mathscr{D}(x;\vec{}[\hat{\Theta_s},\Theta_k])$; the transition between states is made according to $G(\tilde{\Theta}_{k+1},\Theta_k)$ in Algorithm 2. For simplicity, it is assumed that the point sets are rigidly aligned and a purely non-rigid component of the deformation is sought-after.

An optimization "level" K is complete when all of the basis function have been "placed." Initially, the means and weights of all bases are trivially initialized to zero; a basis function is said to be "placed" at the current "level" after its parameters have been optimized. For $N = 10$, the first level is complete when $i = 10$ since all bases have been placed, and the K^{th} optimization level begins by optimizing over parameters of the first basis, keeping the rest constant according to the values found during level $K - 1$. For example, Fig. 5(a) shows the registration result when parameters for the first four bases have been optimized while the others remain trivially initialized. At $K = 1$, $i = 10$ in Fig. 5(b), all bases have been optimized. For the second level $K = 2$, Fig. 5(c) shows registration results after parameters for the first six bases have been optimized, one at a time, with respect to the other bases. In this manner, various states continue to be explored until registration is terminated at the end of level three, Fig. 5(f), and alignment is achieved.

V. Implementation Details

A. Parameter Selection

It was sufficient to set bandwidth σ of the Gaussian kernel in Eq. (2) and Eq. (3) according to the following rule

$$\sigma = \min\left(\frac{1}{L}\sum_{i=1}^{L}\min_{j}\|d_i - d_j\|, \frac{1}{K}\sum_{k=1}^{K}\min_{l}\|m_k - m_l\|\right),$$
$$\forall j \in [1{:}L],\ i \neq j \text{ and } \forall l \in [1{:}K],\ k \neq l$$

for all experiments in Section VI. The example in Fig. 6 illustrates the effect σ has on the registration. We use scaled versions of σ to explore the effect of this parameter: $\sigma_1 = \frac{\sigma}{2}$, $\sigma_2 = \sigma$, $\sigma_3 = 4\sigma$. Performance is robust when the kernel width is set to values around σ but the algorithm starts to break down if σ is significantly reduced or increased. The covariance matrix Q_i specifying the noise model u_k in Eq. (15) was chosen to be

$$Q_0 = \mathrm{diag}\left(\left[.01, \frac{\pi}{90}, \frac{\pi}{90}, \frac{\pi}{90}, .01, .01, .01\right]\right)$$

as the covariance matrix for optimizing the similarity transformation parameters and

$$Q_i = \mathrm{diag}([.01, .01, .01, .005, .005, .005, .001])$$

as the covariance matrix for GRBFs' variables, where $i \in [1,N]$. These choices assume that data is scaled to lie in the unit cube, and they were used for all experiments.

B. Evaluating the Injectivity Constraint

Smoothness of a deformation can be enforced by checking that, in the domain S, $\forall x \vec{\in} S$ the condition

$$G_{lb} < d_J(\vec{x}) \quad (38)$$

holds. If the greatest lower bound of $d_J(\vec{x})$ is $G_{lb} = 0$, from Eq. (24), the deformation is injective everywhere. A smoother deformation can be found by setting G_{lb} to a value greater than zero and requiring that Eq. (38) holds. A related problem was considered in [7]: finding all of the maxima of a Gaussian mixture. In this work, the minimum of the derivative of a Gaussian mixture is desired for checking Eq. (38), which can be seen from Eq. (40). Furthermore, the global minimum only is required.

The scaled, rigid transformation in Eq. (6) can be represented as a matrix

$$A = \begin{bmatrix} s\cdot\cos(\theta) & -s\cdot\sin(\theta) \\ s\cdot\sin(\theta) & s\cdot\cos(\theta) \end{bmatrix}. \quad (39)$$

Then, for the two dimensional case,

$$
\begin{aligned}
d_J(\vec{x}) &= \frac{\partial \mathscr{L}_x}{\partial x}\frac{\partial \mathscr{L}_y}{\partial y} - \frac{\partial \mathscr{L}_x}{\partial y}\frac{\partial \mathscr{L}_y}{\partial x} \\
\frac{\partial \mathscr{L}_x}{\partial x} &= s\cdot\cos(\theta) - \sum_{i=1}^{N}\frac{w_{x_i}}{\sigma_i^2}(x-\mu_{x_i})e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2} \\
\frac{\partial \mathscr{L}_x}{\partial y} &= s\cdot\sin(\theta) - \sum_{i=1}^{N}\frac{w_{x_i}}{\sigma_i^2}(y-\mu_{y_i})e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}
\end{aligned}
\tag{40}
$$

and $\frac{\partial \mathscr{L}_y}{\partial x}$, $\frac{\partial \mathscr{L}_y}{\partial y}$ can be computed similarly. An analytic solution for the minimum of $d_J(\vec{x})$ in Eq. (40) does not exist. Instead, a numerical approach for finding the minimum is presented. While the 2D case is described in detail, the algorithm similarly extends to higher dimensions.

Minimization of $d_J(\vec{x})$ is presented in Algorithm 3: it is performed iteratively by dividing an initial domain S and bounding the function over the subdomains. From the symmetry of Gaussian basis functions, it is known that the minimum of the mixture will lie in the convex hull of the maxima of the individual Gaussian partial derivatives that compose the mixtures in Eq. (40).

Algorithm 3

Algorithm to bound $det(J(\mathscr{L}(\vec{x};\vec{\Theta})))$

1:	**function** TEST_GLB(Θ, ε,G_{lb})	▷ find the greatest lower bound of $d_J(\vec{x})$
2:	S = {Ω_0}	▷ Ω_0 = convex hull of all derivative of Gaussian peaks
3:	bdRange = 2ε	
4:	**while** bdRange > ε **do**	▷ Run until can approximate det() within ε everywhere
5:	N = numberOfElements(S)	
6:	**for** i=1:N **do**	
7:	$\Omega_c = S(i)$	
8:	$S = S \setminus \{\Omega_c\}$	▷ Remove Ω_c from the list
9:	$[U_{\Omega_i}, L_{\Omega_i}]$ = computeBounds(Ω_c,Θ)	
10:	**if** (U_{Ω_i} <= G_{lb}) | (($L_{\Omega_i} \le G_{lb}$) & ($U_{\Omega_i} - L_{\Omega_i} \le \varepsilon$)) **then**	
11:	return 1	▷ Zero crossing detected or within ε of crossing
12:	**else**	
13:	$[\Omega_{N+1}, ...,\Omega_{N+4}]$ = splitDomainUniformly(Ω_c)	▷ Split into 4 regions
14:	S = {S,Ω_{N+1}, ...,Ω_{N+4}}	▷ Add to the list
15:	$\text{bdRange} = \max_{\Omega \in S}(U_\Omega - L_\Omega)$	
16:	**if** $S == \emptyset$ **then**	▷ No subdomain contains zero crossing, det() > 0
17:	return 0	

The algorithm initializes the list S containing *subdomains of interest* to the convex hull Ω_0 of all maxima/minima of the components making up the mixtures in Eq. (40). In line 9, the upper bound U_{Ω_i} and lower bound L_{Ω_i} for d_J are computed for each subdomain in S. If the current sudomain Ω_c cannot be ruled out (*i.e.*, Eq. (38) may be violated in Ω_c), Ω_c is subdivided evenly into four subdomains, line 13, for further consideration. If a subdomain is found to violate Eq. (38) or be within ε of doing so, the function exits with a *true* value.

Otherwise, the test terminates with a *false* value in two cases: S is empty, line 16, and no violations were found, or no violations are found and d_J has been bounded within ε, line 15.

Bounding d_J by U_{Ω_c} and L_{Ω_c} over $\Omega_c \subset \Omega_0$ involves making several conservative estimates. First, we bound

$$\begin{gathered} \min_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_x}{\partial x}\frac{\partial\mathscr{L}_y}{\partial y}\right)-\max_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_x}{\partial y}\frac{\partial\mathscr{L}_y}{\partial x}\right) \\ \leq d_J(\overrightarrow{x}) \leq \\ \max_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_x}{\partial x}\frac{\partial\mathscr{L}_y}{\partial y}\right)-\min_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_x}{\partial y}\frac{\partial\mathscr{L}_y}{\partial x}\right) \quad \text{and} \end{gathered} \tag{41}$$

$$\min_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_x}{\partial x}\frac{\partial\mathscr{L}_y}{\partial y}\right)=\min\{ac,ad,bc,bd\} \text{ where} \tag{42}$$

$$a=\min_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_x}{\partial x}\right) \quad b=\max_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_x}{\partial x}\right) \text{ and} \tag{43}$$

$$c=\min_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_y}{\partial y}\right) \quad d=\max_{\overrightarrow{x}\in\Omega_c}\left(\frac{\partial\mathscr{L}_y}{\partial y}\right). \tag{44}$$

The remaining three parts of Eq. (41) are computed similarly to Eq. (42). Then, the partial derivatives $\frac{\partial\mathscr{L}_x}{\partial x}, \frac{\partial\mathscr{L}_y}{\partial y}, \frac{\partial\mathscr{L}_x}{\partial y}, \frac{\partial\mathscr{L}_y}{\partial x}$ are bounded to find $\min_{\overrightarrow{x}\in\Omega_c}(\cdot)$ and $\min_{\overrightarrow{x}\in\Omega_c}(\cdot)$ in Eq. (42)–Eq. (44):

$$\begin{gathered} s\cdot\cos\theta-\sum_{i=1}^{N}\max_{\overrightarrow{x}\in\Omega_c}\left(\frac{w_{x_i}}{{\sigma_i}^2}(x-\mu_{x_i})\cdot e^{-\frac{1}{2\sigma_i^2}\|\overrightarrow{x}-\mu_i\|^2}\right) \\ \leq \frac{\partial\mathscr{L}_x}{\partial x} \leq \\ s\cdot\cos\theta-\sum_{i=1}^{N}\min_{\overrightarrow{x}\in\Omega_c}\left(\frac{w_{x_i}}{{\sigma_i}^2}(x-\mu_{x_i})\cdot e^{-\frac{1}{2\sigma_i^2}\|\overrightarrow{x}-\mu_i\|^2}\right). \end{gathered} \tag{45}$$

Further, the value $\min_{\overrightarrow{x}\in\Omega_c}(\cdot)$ in Eq. (45) is computed as

$$\begin{gathered} \min_{\overrightarrow{x}\in\Omega_c}\left(\frac{w_{x_i}}{{\sigma_i}^2}(x-\mu_{x_i})\cdot e^{-\frac{1}{2\sigma_i^2}\|\overrightarrow{x}-\mu_i\|^2}\right) \\ =\begin{cases} b_1 & \text{if } \min_{\overrightarrow{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i}) \leq 0 \\ b_2 & \text{if } \min_{\overrightarrow{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i}) > 0 \end{cases} \\ b_1=\frac{1}{{\sigma_i}^2}\min_{\overrightarrow{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i})\cdot\max_{\overrightarrow{x}\in\Omega_c} e^{-\frac{1}{2\sigma_i^2}\|\overrightarrow{x}-\mu_i\|^2} \\ b_2=\frac{1}{{\sigma_i}^2}\min_{\overrightarrow{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i})\cdot\min_{\overrightarrow{x}\in\Omega_c} e^{-\frac{1}{2\sigma_i^2}\|\overrightarrow{x}-\mu_i\|^2} \end{gathered} \tag{46}$$

and the computation is similar for $\max_{x\in\Omega_c}(\cdot)$ of Eq. (45). The bounds on $\min_{\vec{x}\in\Omega_c}(\cdot)$ of Eq. (46) (similarly on $\max_{\vec{x}\in\Omega_c}(\cdot)$) are

$$\min_{\vec{x}\in\Omega_c} w_{x_i}(x-\mu_{x_i}) = \begin{cases} w_{x_i}\max_{\vec{x}\in\Omega_c}(x-\mu_{x_i}), & w_{x_i}\leq 0 \\ w_{x_i}\min_{\vec{x}\in\Omega_c}(x-\mu_{x_i}), & w_{x_i}> 0. \end{cases} \quad (47)$$

To determine the bounds for $\min_{\vec{x}\in\Omega_c} e^{(\cdot)}$ and $\max_{\vec{x}\in\Omega_c} e^{(\cdot)}$ in Eq. (46) and compute Eq. (47), the domain is broken into zones as shown in Fig. 7; the corner points $P_j = [P_{j_x}, P_{j_y}]$, $j \in \{1, ..., 4\}$ determine the boundaries. Now, the quantities in Eq. (47) can be bounded as

$$\begin{aligned} \min_{\vec{x}\in\Omega_c}(x-\mu_{x_i}) &\geq P_{1_x}-\mu_{x_i} \\ \max_{\vec{x}\in\Omega_c}(x-\mu_{x_i}) &\leq P_{2_x}-\mu_{x_i} \\ \min_{\vec{x}\in\Omega_c}(y-\mu_{y_i}) &\geq P_{3_y}-\mu_{y_i} \\ \max_{\vec{x}\in\Omega_c}(y-\mu_{y_i}) &\leq P_{1_y}-\mu_{y_i}. \end{aligned} \quad (48)$$

Finally, bounds on the Gaussian functions within Ω_c are computed using Table I. This table is made by examining Fig. 7(a). For example, if a Gaussian function is located in Zone 3, its maximum value is located at the boundary of the green line and Ω_c and the minimum is either at P_3 or P_4. All possible scenarios for the location of a Gaussian function and the corresponding bounds are listed in Table I.

VI. Results and Experiments

A. Examples in 2D

a) Bhattacharyya Registration—In this example, we show that the stochastic optimization method can be applied to a variety of cost functionals for registration. The Bhattacharyya distance [4] measures how similar two PDFs are to each other. It is defined as

$$D_b(p,q) = -\ln(B_c(p,q)) \quad \text{where} \quad (49)$$

$$B_c(p,q) = \int \sqrt{p(x)q(x)}dx \quad (50)$$

and $p(x)$, $q(x)$ are the two PDFs being compared. The Bhattacharyya coefficient lies in the interval $0 \leq B_c \leq 1$, and values closer to 1 indicate that PDFs are more similar. In the case of point set registration, we have two probability distributions: $m(x;\mathcal{M})$ and $c(x;\mathcal{Y}(D))$ from Eq. (2)–(3). Registration is achieved by minimizing the energy

$$E_B(M, \mathcal{Y}(D)) = 1 - B_c(m(\vec{x}, M), c(\vec{x}; \mathcal{Y}(D))). \quad (51)$$

A registration result using the Bhattacharyya energy defined by Eq. 51 is presented in Fig. 8. The optimization process, described in Section IV-B, is illustrated by showing energy values at each stage of the optimization. All other experiments in this paper are based on the energy defined in Eq. (4).

b) Noisy and Incomplete Data 2D—The first set of tests demonstrate the ability of the proposed algorithm to handle missing points where no correspondences exist and noisy data sets, with additional points not belonging to the structures of interest. The ground truth data was generated by taking a model point set (from http://www.cise.ufl.edu/~anand/students/chui/research.html) and deforming it to create an ideal data set, both depicted in Fig. 2(a). The test data was created by adding varying levels of noise as presented in Fig. 10(a) or by removing points from the two sets as in Fig. 10(d).

The baseline algorithms used for comparison are: CPD [32], RPS [24], and RPM [10] (see Section II). For RPS, the authors proposed two parameterizations for the deformation: TPS and GRBFs, labeled 'TPS L2' and 'GRBF L2', respectively, in Figs. 9(a) and 9(b). For CPD, the user selects values for two parameters: the regularization weight λ and the Gaussian basis width β. For RPS using TPS, the user sets λ and σ, the width of the kernel in Eq. (2)–(3). For RPS using GRBFs, λ, β, σ must be chosen. Finally, for RPM the initial temperature T_0 and the annealing rate r must be set. In our experience, these parameters had to be adjusted for each noise level and missing point level to obtain the best registration. Since it is not obvious what the settings should be, a parameter sweep was performed for each of the competing approaches using 56 parameter pairs. 'GRBF L2' has three parameters, so 392 parameter triplets were tried per registration. Thus, the best possible results are reported in Fig. 9 for the competing approaches. Outside of the constants set in Section V-A and the number of particle and iterations for the particle filter (defined once and used for all experiments), the stochastic point set registration (SPSR) algorithm has just one parameter: the number of basis functions. The user only has access to this parameter and changing it has the intuitive effect of improving registration accuracy, explained in Section IV-C, at the cost of increased computation time. For all experiments, $N = 10$, which was empirically determined as reasonable compromise between running time and accuracy.

The plot of error as a function of noise level, depicted in Fig. 9(a), was generated by creating ten synthetic examples at each noise level plotted, performing the registration, then plotting the average error rate of the ten trials. Similarly, the error as a function of the missing points ratio, in Fig. 9(b), was generated by removing a certain percentage of points from each data set and computing the registration error. Given the ground truth, the error is computable by taking the average of the Euclidean distance between all points and their correspondences in the stationary set (error vectors are shown as green arrows in Fig. 10(b) and Fig. 10(e)).

The SPSR performs favorably compared to the competing algorithms in both tests while at the same time maintaining a non-overlapping deformation field Eq. (24) (not guaranteed by any of the competing approaches). In a real application, the competing approaches will likely not achieve the theoretical performance shown in Fig. 9 because ground truth is not available, and requiring a human to perform a parameter sweep and select the best of 56 registrations is time consuming and is not trivial.

c) Landmark Constrained 2D Example—This section demonstrates an example in which user constraints greatly simplify the registration problem. The original, misaligned point sets are depicted in Fig. 11(a). Two maps were created by projecting the earth's surface onto a plane according to the Lambert and Mercator projections. The resulting shape of the South American continent differs widely; to create the point sets, images of South America, from the two projections, were sampled uniformly. Additionally, the coordinates of the following four cities, denoted by pentagons and squares, were marked in each image: Brasilia, Buenos Aires, Manaus, and Ushuaia. Thus, the registration problem consists of aligning the point sets subject to the condition that location of the cities match exactly.

The first step of registration establishes a rough alignment using just the four pairs of corresponding cities. Using the known correspondences, an optimal interpolating thin plate spline (TPS) was computed. Then, the point sets were aligned using the TPS deformation field. The results in Fig. 11(b) clearly show that four correspondences are insufficient to align the point set exactly; however, the gross alignment errors have been corrected. At this point, it is known that the location of the cities match; consequently, using the approach in Section IV-A these points are fixed for the remainder of the registration process. Subject to these constraints, the SPSR algorithm is applied to achieve the final registration in Fig. 11(b).

B. Examples in 3D

a) Synthetic Example—The extension of SPSR to 3D is straightforward; it involves a mild increase in the optimization space of three degrees of freedom for the rigid parameters and two degrees of freedom per Gaussian basis. Since the increase is moderate, we expect similar performance in 3D as for 2D. In the first experiment, a point cloud of an elephant was generated with 3,093 points and was deformed to create the two point clouds in Fig. 12(a). The deformation is large in magnitude and twists the trunk of the elephant in 3D. The local nature of the transformation was intentional. We see that the SPSR algorithm correctly focuses on the region around the trunk and only deforms space in this region without altering far away areas.

b) Real 3D Example—For the second experiment, computed tomography (CT) image volumes of two different patients were obtained. With a simple threshold, bone label maps were extracted; these labels were then uniformly sampled to generate point clouds consisting of 14,832 and 14,723 points. The initial point sets are shown in Fig. 13(a), and despite a rigid alignment, significant differences exist. Here, the target application is atlas based segmentation (i.e., align the CT volumes using just the sampled point clouds and transfer label maps from the reference patient to the incoming patient). The results of registration are shown in Fig. 13. In this example, ground truth label maps for the mandible, larynx, and spinal cord are available for both patients, in Fig. 13(e). Using the computed deformation fields from the point set registration, the label maps of the reference image were deformed. Clearly, in Fig. 13(f), the organ labels are well aligned.

c) Landmark Constrained 3D Example—Despite accurate registration of the bony structures and organs around them, in areas where no information was available (i.e., no

points present to guide the registration), there is no guarantee that the original CT image will be aligned. A second stage of registration is performed to align these regions by first warping the initial CT volume according to the deformation computed in Fig. 13. At this stage, a label map of the patients' flesh was made, again using a simple threshold operation. The point clouds that are a result of sampling the masks are shown in Fig. 14(a) and Fig. 14(b).

This problem is complicated by the fact that registering the flesh point clouds would likely undo the alignment of the bone achieved in Fig. 13. We can mitigate this complication by introducing a set of constraints, as in Section IV-A. The constraint points are chosen as a subset of the aligned skeleton point clouds and do not move at all during the subsequent registration. Thus, regions belonging to the bone tissue are restricted in their motion. To generated the deformation field in Fig. 14, 67 points serve to constrain the problem. Subject to the constraints, registration is performed and the deformation field in Fig. 14(c) and Fig. 14(d) are computed. Finally, the originally misaligned CT volumes from Fig. 14(e) are registered using a composition of the two deformation fields from Fig. 13–14 to obtain the registration visualized in Fig. 14(f).

C. Performance

The experiments were all done on an Intel Core i7-2600K processor. For the smaller, 2D examples in Fig. 9, the computations were completed in 10 minutes. The most computationally intensive example was the landmark constrained registration in Fig. 14; this example took 16 hours to compute. There was no attempt to optimize the algorithm's performance. Significant performance improvements can be achieved by doing the computations in parallel. The proposed framework is perfectly parallelizable: each particle can be computed independently. Making the computations in parallel, on a graphics processing unit(GPU), for example, is expected to result in roughly a two orders of magnitude speed improvement.

VII. Conclusion, Discussion, Future Work

This work presented a stochastic registration methodology and demonstrated it on registration of point sets. The resulting registration framework was robust for scenes with noise and missing points; it can be applied to points clouds of arbitrary dimension. The deformation field $\mathscr{L}$ consisted of the additive composition of a rigid transformation and a non-rigid transformation through a GRBF network. The key parameter, N (the number of basis functions), can be increased to improve the registration with increased run time being the only detrimental effect. The effect of increasing N is intuitive since it simply allows the algorithm to treat increasingly finer deformations as opposed to a smoothness penalty employed by state of the art point set registration algorithms (e.g., RPS and CPD). Additionally, important constraints on the transformation $\mathscr{L}$ are imposed to ensure physically realistic deformations. This last point is essential for medical imaging and other applications that obtain point sets from physical objects. Further work includes experimenting with cost functions and constraints to expand the applicability of the proposed framework.

Acknowledgments

This project was supported by in part by grants from the National Center for Research Resources (P41-RR-013218) and the National Institute of Biomedical Imaging and Bioengineering (P41-EB-015902) of the National Institutes of Health. This work was also supported by NIH grants R01 MH82918 and 1U24CA18092401A1 as well as AFOSR grants FA9550-12-1-0319 and FA9550-15-1-0045.

References

1. Andrieu C, de Freitas N, Doucet A, Jordan M. An Introduction to MCMC for Machine Learning. Machine Learning. 2003; 50(1):5–43.
2. Avants B, Epstein C, Grossman M, Gee J. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. Medical Image Analysis. 2008:26–41. [PubMed: 17659998]
3. Besl P, McKay H. A method for registration of 3-d shapes. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on. Feb; 1992 14(2):239–256.
4. Bhattacharyya A. On a measure of divergence between two statistical populations defined by their probability distributions. Bulletin of Cal Math Soc. 1943; 35(1):99–109.
5. Bookstein FL. Principal warps: Thin-plate splines and the decomposition of deformations. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on. 1989; 11(6):567–585.
6. Brown M, Lowe D. Automatic panoramic image stitching using invariant features. International Journal of Computer Vision (IJCV). 2007; 74(1):59–73.
7. Carreira-Perpinan M. Mode-finding for mixture of gaussian distributions. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on. 2000; 22(11):1318–1323.
8. Chen, Z. Tech rep. McMaster University; 2003. Bayesian filtering: From kalman filters to particle filters, and beyond.
9. Chui, H.; Rangarajan, A. A feature registration framework using mixture models. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA), Proceedings of the; 2000. p. 190-197.
10. Chui H, Rangarajan A. A new point matching algorithm for non-rigid registration. Computer Vision and Image Understanding. 2003; 89(2–3):114–141.
11. Del Moral P, Doucet A, Jasra A. Sequential Monte Carlo samplers. Journal of the Royal Statistical Society: Series B. 2006; 68(3):411–436.
12. van den Elsen, P.; Pol, E.; Viergever, M. Engineering in Medicine and Biology Magazine. Vol. 12. IEEE; Mar. 1993 Medical image matching-a review with classification; p. 26-39.
13. Fornefett M, Rohr K, Stiehl H. Radial basis functions with compact support for elastic registration of medical images. Image and Vision Computing. 2001; 19(12):87–96.
14. Glaunes, J.; Trouve, A.; Younes, L. Diffeomorphic matching of distributions: a new approach for unlabelled point-sets and sub-manifolds matching. Computer Vision and Pattern Recognition (CVPR), Proceedings of the 2004 IEEE Computer Society Conference on; June 2004; p. II–712-II–718.
15. Gordon N, Salmond D, Smith A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. Radar and Signal Processing, IEE Proceedings F. 1993; 140:107–113.
16. Grenander U, Srivastava A, Saini S. A pattern-theoretic characterization of biological growth. Medical Imaging(TMI), IEEE Transactions on. May; 2007 26(5):648–659.
17. Guo, H.; Rangarajan, A.; Joshi, S. Diffeomorphic point matching. In: Paragios, N.; Chen, Y.; Faugeras, O., editors. Handbook of Mathematical Models in Computer Vision. Springer US; 2006. p. 205-219.
18. Guo H, Rangarajan A, Joshi S, Younes L. Non-rigid registration of shapes via diffeomorphic point matching. Biomedical Imaging: Nano to Macro, IEEE International Symposium on. Apr.2004 1:924–927.
19. Wahba, G., editor. Spline Models for Observational Data. SIAM; Philadelphia, PA: 1990.
20. Harris C, Stephens M. A Combined Corner and Edge Detection. Alvey Vision Conference, Proceedings of the. 1988:147–151.

21. Izard, C.; Jedynak, B. Bayesian registration for anatomical landmark detection. Biomedical Imaging: Nano to Macro. IEEE International Symposium on; April 2006; p. 856-859.
22. Ji, C.; Zhang, Y.; Tong, M.; Yang, S. Particle filter with swarm move for optimization. International conference on Parallel Problem Solving from Nature (PPSN), Proceedings of the; Berlin, Heidelberg: Springer-Verlag; 2008. p. 909-918.
23. Jian, B.; Vemuri, B. A robust algorithm for point set registration using mixture of gaussians. Vol. 2. IEEE Computer Society; Los Alamitos, CA, USA: 2005. p. 1246-1251.
24. Jian B, Vemuri B. Robust point set registration using gaussian mixture models. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on. Aug; 2011 33(8):1633–1645.
25. Jung, H.; Lee, K.; Lee, S. Window annealing over square lattice markov random field. Proceedings of the 10th European Conference on Computer Vision: Part II; European Conference on Computer Vision (ECCV), Proceedings of the, Springer-Verlag; Berlin, Heidelberg. 2008. p. 307-320.
26. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983; 220(4598):671–680. [PubMed: 17813860]
27. Kurtek S, Srivastava A, Klassen E, Laga H. Landmark-guided elastic shape analysis of spherically-parameterized surfaces. Computer Graphics Forum. 2013; 32(2):429–438.
28. Locatelli M. Simulated annealing algorithms for continuous global optimization: convergence conditions. Journal of Optimization Theory and Applications. Jan.2000 104:121–133.
29. Loeckx, D.; Maes, F.; Vandermeulen, D.; Suetens, P. Nonrigid Image Registration Using Free-Form Deformations with a Local Rigidity Constraint. International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the, Lecture Notes in Computer Science; Berlin / Heidelberg: Springer; 2004. p. 639-646.chap. 78
30. Ma, B.; Ellis, Y. Surface-based registration with a particle filter. International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Proceedings of the; 2004. p. 566-573.
31. Meisters G, Olech C. Locally one-to-one mappings and a classical theorem on schlicht functions. Duke Mathematics Journal. 1963; 30:63–80.
32. Myronenko A, Song X. Point Set Registration: Coherent Point Drift. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on. 2010; 32(12):2262–2275.
33. Oleskiw, T.; Elder, J.; Peyre, G. On growth and formlets: Sparse multi-scale coding of planar shape. Computer Vision and Pattern Recognition (CVPR), IEEE Conference on; June 2010; p. 459-466.
34. Park J, Sandberg IW. Universal approximation using radial-basis-function networks. Neural Computation. Jun.1991 3:246–257.
35. Peter A, Rangarajan A. Shape analysis using the fisher-rao riemannian metric: unifying shape representation and deformation. Biomedical Imaging: Nano to Macro, IEEE International Symposium on. Apr.2006 :1164–1167.
36. Rangarajan, A.; Chui, H.; Bookstein, F. The softassign procrustes matching algorithm. International Conference on Information Processing in Medical Imaging, Proceedings of; Springer; 1997. p. 29-42.
37. Robert, C.; Casella, G. Monte Carlo Statistical Methods (Springer Texts in Statistics). Springer-Verlag New York, Inc; Secaucus, NJ, USA: 2005.
38. Rohr K, Stiehl H, Sprengel R, Buzug T, Weese J, Kuhn MH. Landmark-based elastic registration using approximating thin-plate splines. Medical Imaging (TMI), IEEE Transactions on. 2001:526–534.
39. Sandhu R, Dambreville S, Tannenbaum A. Point set registration via particle filtering and stochastic dynamics. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on. 2010; 32:1459–1473.
40. Tsin Y, Kanade T. A correlation-based approach to robust point set registration. European Conference on Computer Vision (ECCV), Proceedings of the. 2004:558–569.
41. Vanderbilt D, Louie S. A monte carlo simulated annealing approach to optimization over continous variables. Journal of Computational Physics. 1984; 56(2):259–271.

42. Vidal C, Jedynak B. Learning to match: Deriving optimal template-matching algorithms from probabilistic image models. International Journal of Computer Vision(IJCV). 2010; 88(2):189–213.
43. Wang F, Vemuri B, Rangarajan A, Eisenschenk S. Simultaneous nonrigid registration of multiple point sets and atlas construction. Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on. 2008; 30(11):2011–2022.
44. Wassermann, D.; Ross, J.; Washko, G.; Westin, C.; San Jose Estepar, R. Diffeomorphic point set registration using non-stationary mixture models. Biomedical Imaging (ISBI), IEEE 10th International Symposium on; April 2013; p. 1042-1045.
45. Zhou, E.; Fu, M.; Marcus, S. A particle filtering framework for randomized optimization algorithms. Winter Simulation Conference, Proceedings of the; WSC '08, Winter Simulation Conference; 2008. p. 647-654.

Biographies

Ivan Kolesov Ivan Kolesov received his Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA. He is a Postdoctoral Research Fellow at Stony Brook University, Stony Brook, NY. His research interests lie in the fields of computer vision, medical imaging, and systems and control.

Jehoon Lee Jehoon Lee received his Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA. He is currently working for Samsung Electronics Co., Ltd., South Korea. His research interests include the usage of linear or nonlinear filters and geometric active contours to solve various problems of segmentation, visual tracking, and pose estimation in computer vision.

Gregory Sharp Greg Sharp received his PhD in Computer Science and Engineering from the University of Michigan in 2002, and completed post-doctoral training in Radiation

Physics from the Massachusetts General Hospital in 2004. Since that time, he has served on the clinical faculty of the department of Radiation Oncology at Massachusetts General Hospital as Assistant Radiation Physicist and has an appointment as Assistant Professor in Harvard Medical School.

Patricio Vela Patricio Vela earned his bachelor of science degree in 1998 and his doctorate in 2003 at the California Institute of Technology, where he did his graduate research on geometric nonlinear control and robotics. After working as a post-doctoral researcher in computer vision at the Georgia Institute of Technology, he joined the School of Electrical and Computer Engineering of the Georgia Institute of Technology in 2005. His research interests lie in the geometric perspectives to control theory and computer vision. Recently, he has been interested in the role that computer vision can play for achieving control-theoretic objectives of (semi-)autonomous systems. His research also covers control of nonlinear systems, typically robotic systems.

Allen Tannenbaum Allen Tannenbaum is Professor of Computer Science and Applied Mathematics/Statistics at Stony Brook University. He does research in systems and control, medical imaging, and computer vision.

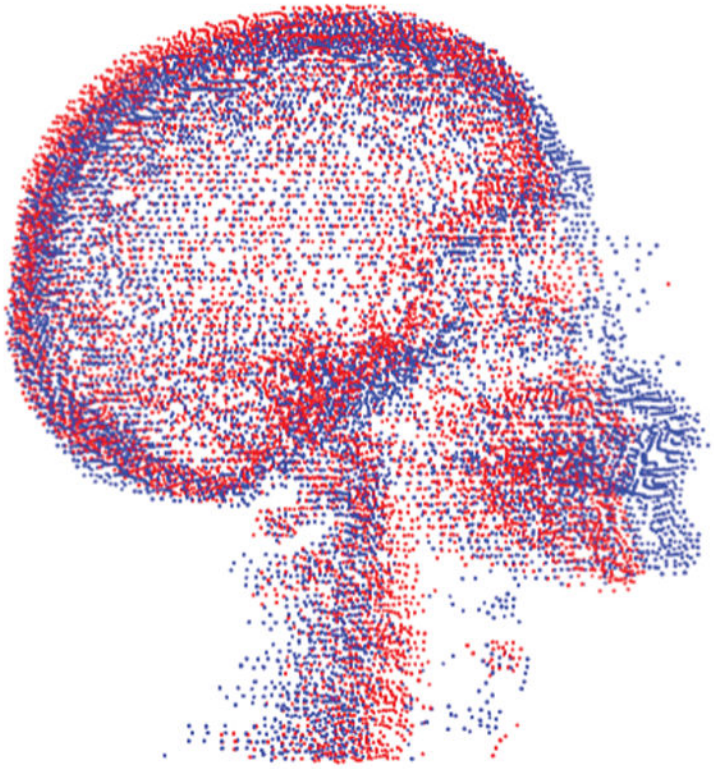
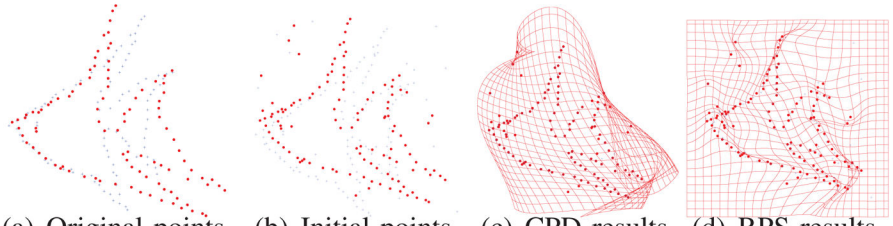


Fig. 1.

Point clouds generated from computed tomography (CT) volumes of two patients are shown. The points are sparse representations of the human skeleton. Any deformation that proposes to relate them must be injective.



(a) Original points. (b) Initial points. (c) CPD results. (d) RPS results.

Fig. 2.

This figure shows the importance of physically realistic non-rigid deformations; registration results for CPD [32] and RPS(GRBF parameterization) [24] when parameter settings do not provide sufficient regularization can be seen in Fig. 2(c) and 2(d), respectively. Although the points match well, the mapping violates fundamental physical principles; these settings, unknown ahead of time, should be avoided.

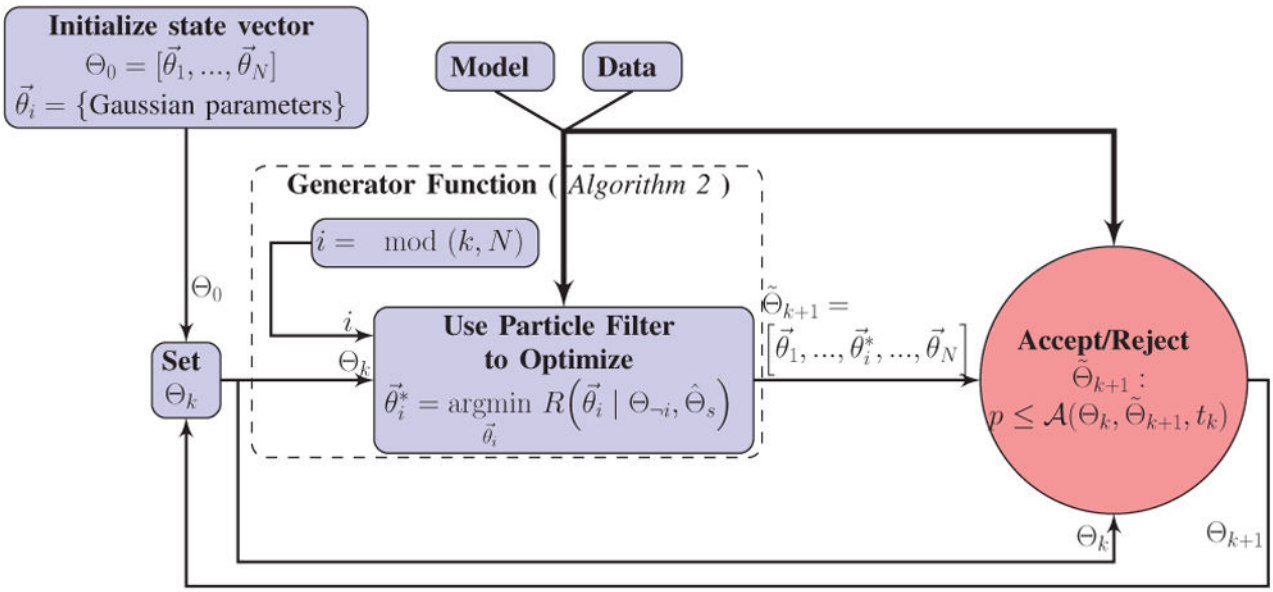


Fig. 3.
The SPSR registration algorithm described in Section IV-B.

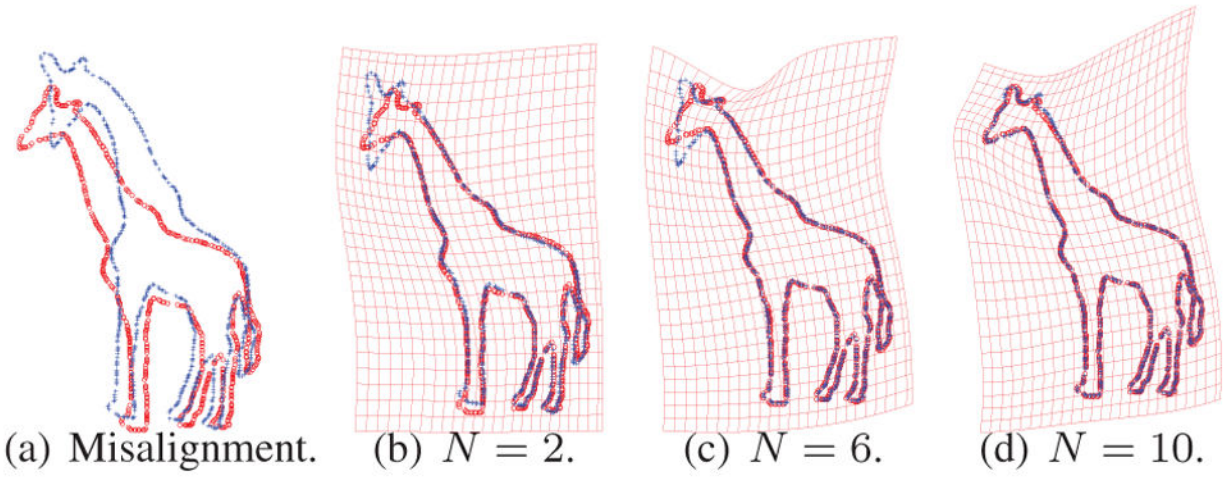


Fig. 4.

Illustration of the effect changing the number of basis functions N in Eq. (6) has on registration results. Starting from the original point sets, Fig. 4(a), the final registration for 2, 6, 10 Gaussian basis functions are show in in Fig. 4(b)–4(d), respectively. Registration accuracy is improved as N increases; however, a larger N provides increasingly marginal improvement. Notice the alignment around the giraffe's head, back, and legs in each figure.

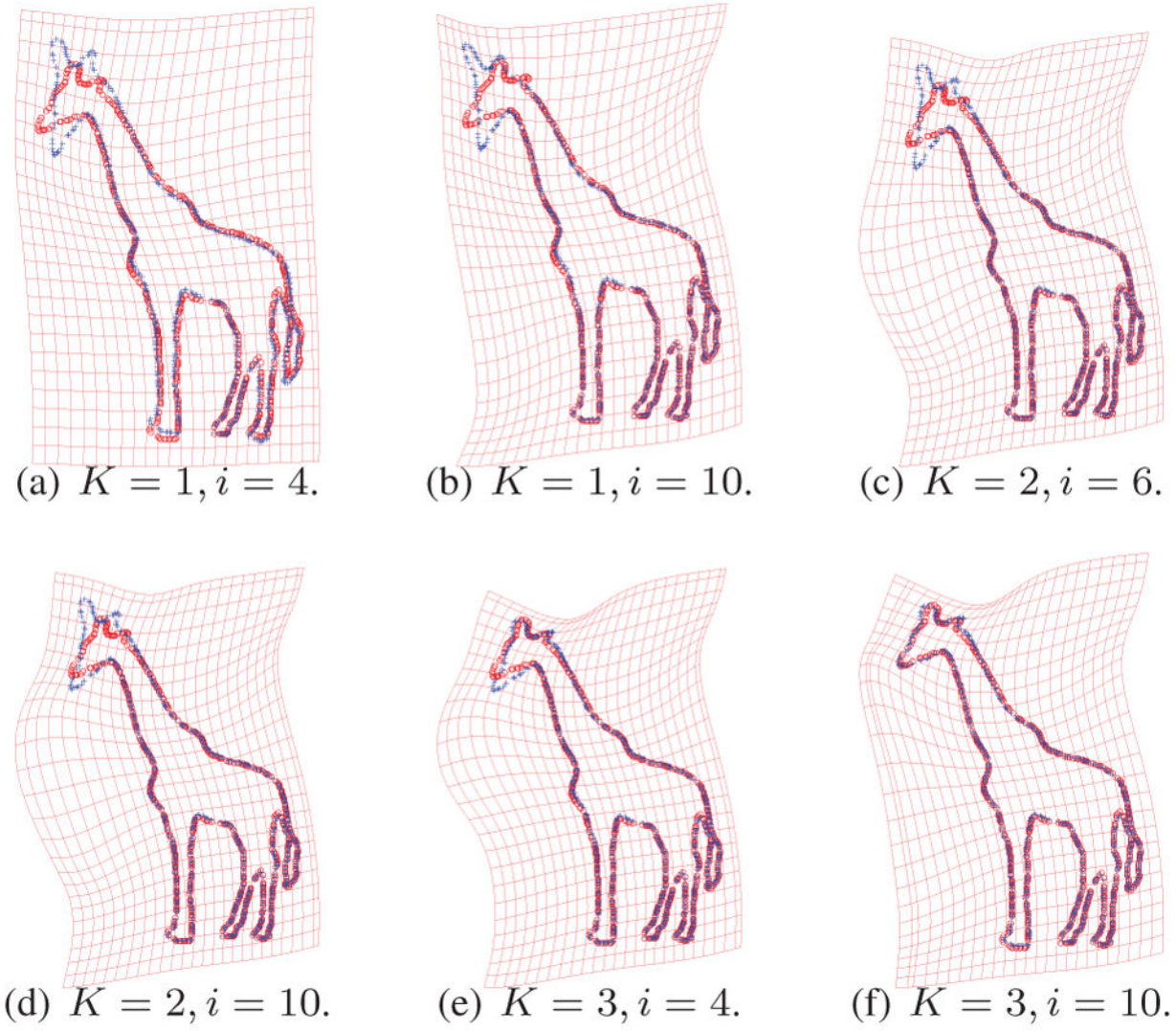


Fig. 5.

The registration process described in Fig. 3 is illustrated. The number of basis functions in Eq. (6) is kept constant at $N = 10$. Each subfigure is the result of an instantiation of the state vector Θ_g.

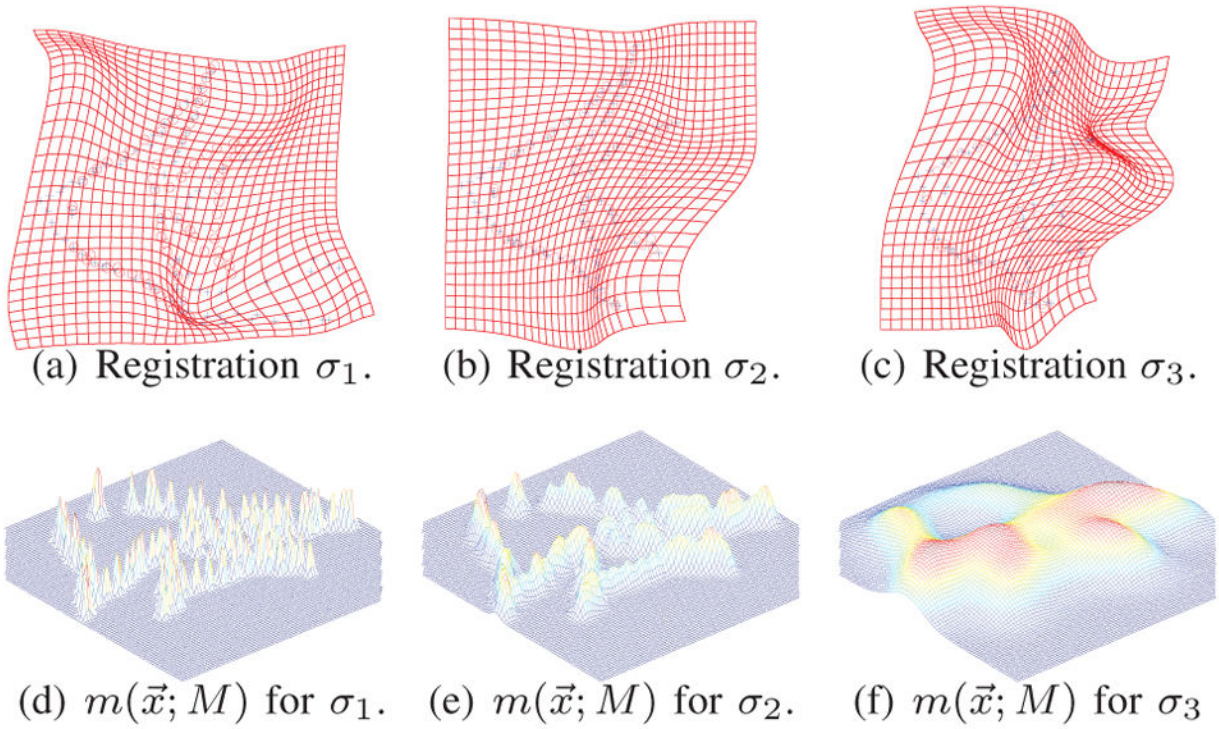
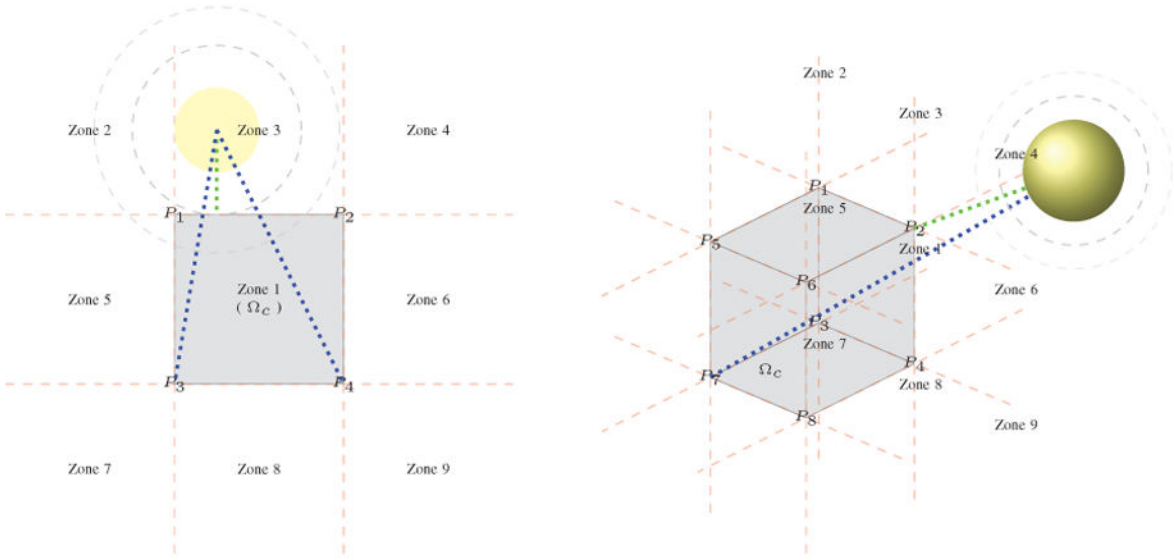


Fig. 6.

Illustration of the effect changing the kernel width in Eq. (2)–(3) has on the registration. The corresponding $m(x;\vec{M})$ is also shown.



(a) Zones in 2D with respect to Ω_c (Zone 1).

(b) Zones in 3D; there are 27 total but only the first 9 are labeled.

Fig. 7.

To find bounds for derivative of Gaussian functions over Ω_c (shown in gray) the 2D/3D space is divided into zones. The circle/sphere show the center of a sample Gaussian function. Clearly, the minimum for this function is located at a point at the end of a blue line and the maximum at the end of the green line.

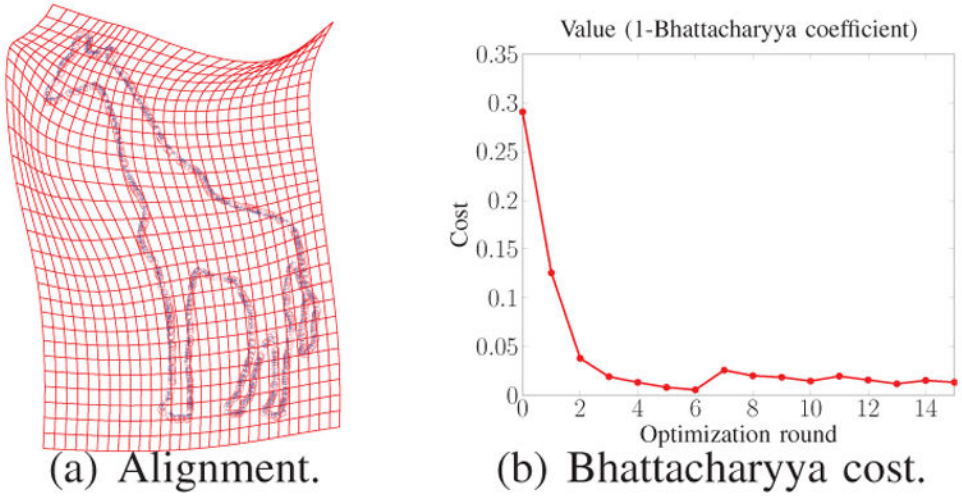
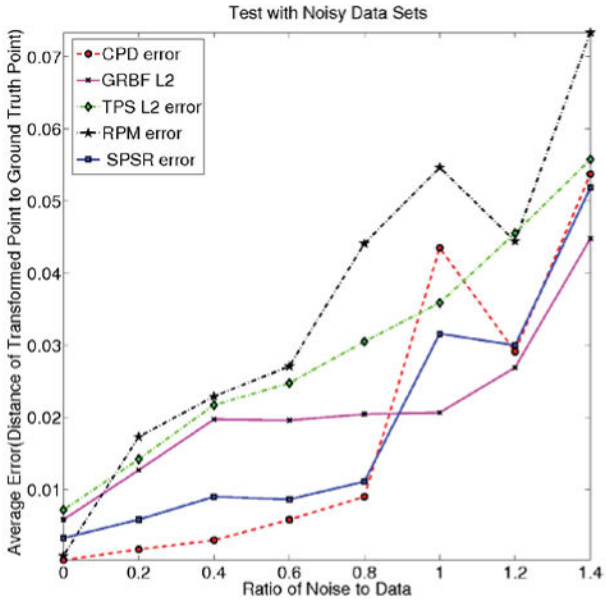
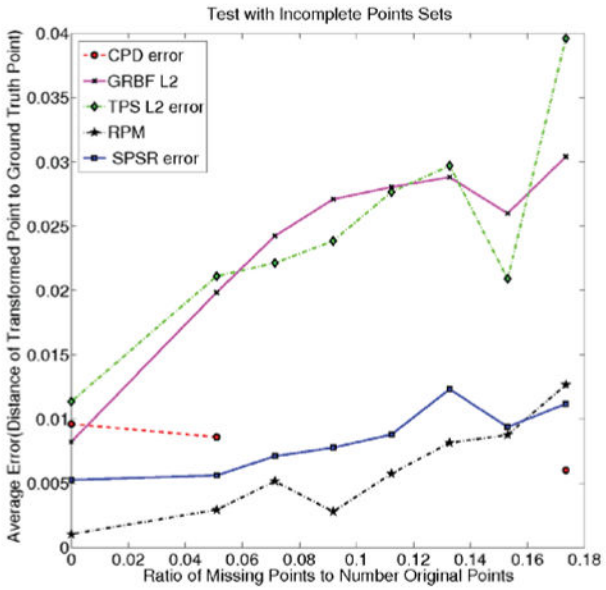


Fig. 8.

Registration is performed by minimizing the Bhattacharyya cost function defined in Eq. 51. Energy values at the end of each optimization round, described in Section IV-B are shown.



(a) Average error per point for noisy data sets.



(b) Average error per point for incomplete data sets.

Fig. 9. Fish Data

Plot of the registration errors for data corrupted by noise and data with missing correspondences; the original data points are shown in Fig. 2(a). SPSR is the approach proposed in this work; its is compared to CPD [32] , RPS [24] (GRBF L2 and TPS L2), and RPM [10]. A parameter sweep is performed for CPD, RPS, and RPM and settings corresponding to the lowest error while producing a one-one mapping are used to generate the graphs. In Fig. 9(b), the graph for CPD has missing data points because it failed to find a non-overlapping mapping for all 56 parameter settings.

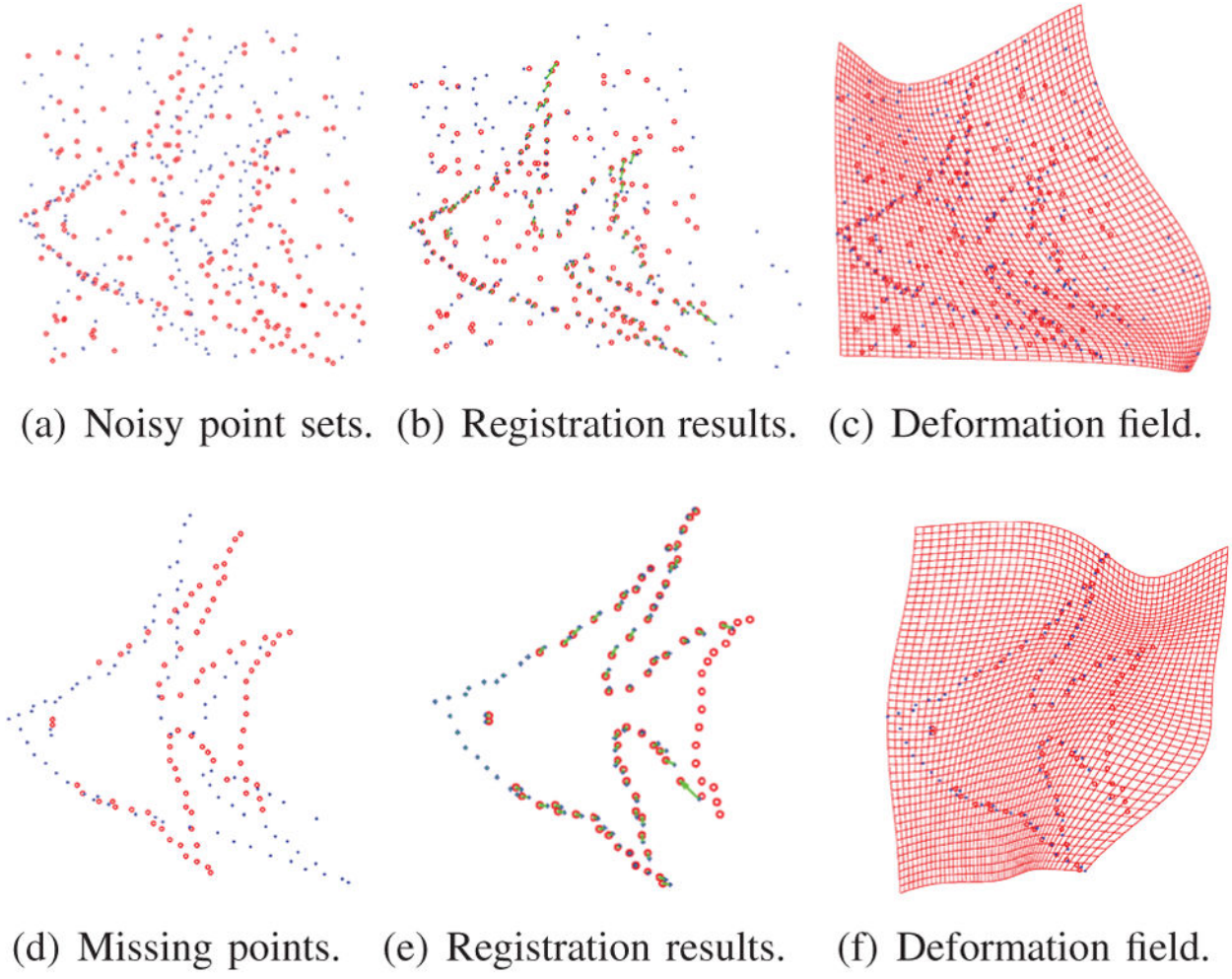


Fig. 10.

The synthetic examples that were used to produce Fig. 9 are shown. The starting point sets are in Fig. 2(a). These sets were corrupted by random noise; Fig. 10(a) shows an example with noise to data ratio of 1.2 . In the other test, one of the sets had points removed from the front and the other from the back; Fig. 10(d) shows an example where 18% of points were removed from each set. Registration results are shown and the errors we measure represented by green arrows.

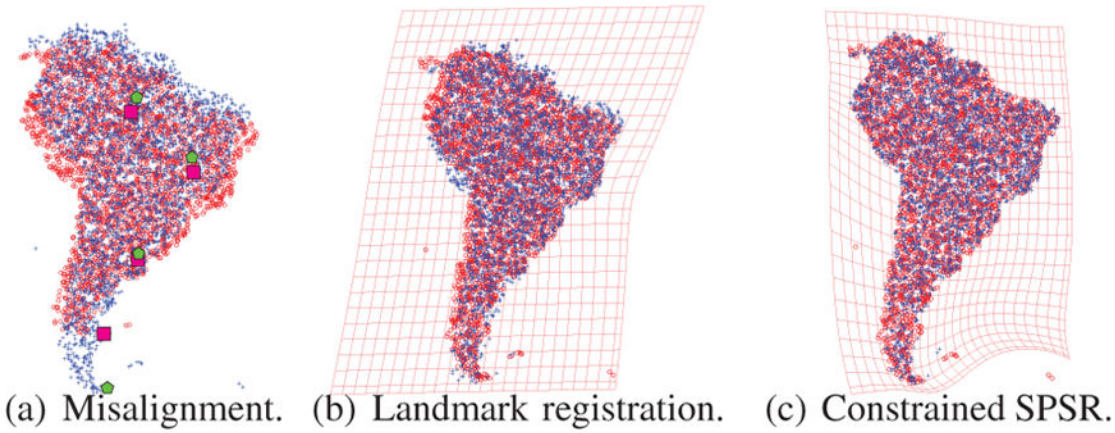
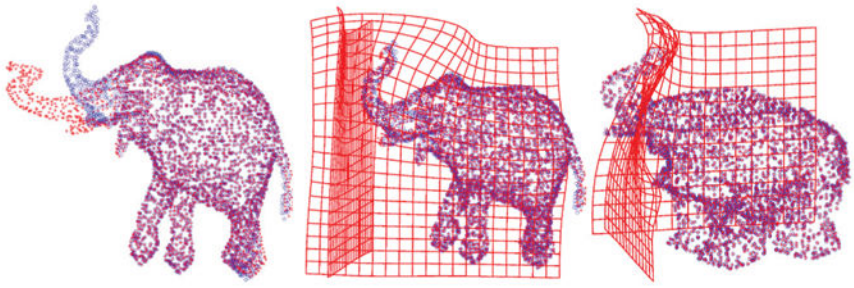


Fig. 11.

An example of constrained registration in 2D is shown here. Two misaligned point sets, along with four corresponding pairs of points are seen in Fig. 11(a). A rough alignment is performed using TPS and the four pairs of corresponding points, shown in Fig. 11(b). The landmarks are aligned and constrained not to move during the final step. Results of doing point set registration subject to known points remaining stationary are in Fig. 11(c).



(a) Elephant point sets. (b) Registration view 1. (c) Registration view 2.

Fig. 12.

A 3D point cloud of an elephant was deformed by applying a known deformation to create a synthetic data set. The sets are composed of 3,093 points. In Fig. 12(b) and 12(c), the computed deformations are not self-overlapping and are local around the misaligned points, as desired.

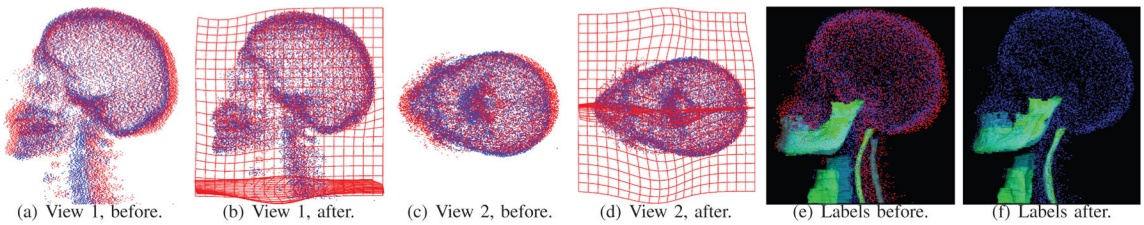
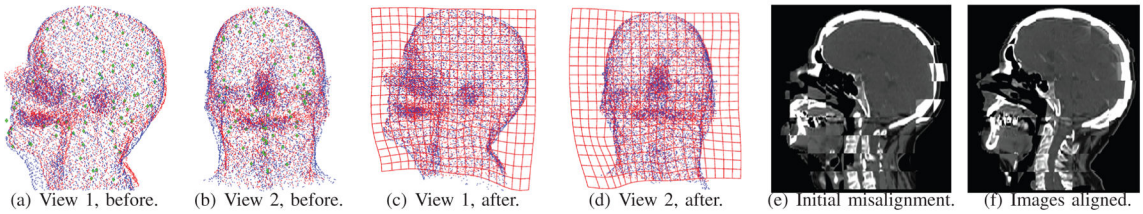


Fig. 13.

Real point clouds (14,832 and 14,723 points) generated from CT volumes of different patients. Clearly, a rigid registration is insufficient as there is large variation in the tilt of the heads (vertebral columns don't align), the shape of the skull, and the shape of the jaws. Medical image registration is the targeted application, and a non-overlapping field is particularly important.



(a) View 1, before. (b) View 2, before. (c) View 1, after. (d) View 2, after. (e) Initial misalignment. (f) Images aligned.

Fig. 14.

Point clouds of 6,842 and 6,877 points were generated from CT volumes by sampling the label maps of patients' flesh. Constrained registration was performed to align the point clouds representing flesh while keeping points on the registered skeletons from Fig. 13 stationary. The original misalignment and the 67 constraint points marked with green diamonds are displayed in Fig. 14(a) and Fig. 14(b). Finally, the deformations from Fig. 13 and Fig. 14 were sequentially applied to the original CT volume. The starting misalignment is seen in Fig. 14(e) and the result of the two step registration is in Fig. 14(f).

Table I

Bounds on a Gaussian function over the domain Ω_c in 2D. The bounds depend on which zone relative to Ω_c , from Fig. 7(a), the function is located in.

Zone	$\min_{x\in\Omega}\left(e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}\right)$	$\max_{x\in\Omega}\left(e^{-\frac{1}{2\sigma_i^2}\|\vec{x}-\mu_i\|^2}\right)$
1	$\min_{j\in\{1,\ldots,4\}} e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2}$	1
2	$e^{-\frac{1}{2\sigma_i^2}\|P_4-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_1-\mu_i\|^2}$
3	$\min_{j\in\{3,4\}} e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|\left[0,P_{1_y}\right]-\mu_i\|^2}$
4	$e^{-\frac{1}{2\sigma_i^2}\|P_3-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_2-\mu_i\|^2}$
5	$\min_{j\in\{2,4\}} e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|[P_{1_x},0]-\mu_i\|^2}$
6	$\min_{j\in\{1,3\}} e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|[P_{2_x},0]-\mu_i\|^2}$
7	$e^{-\frac{1}{2\sigma_i^2}\|P_2-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_3-\mu_i\|^2}$
8	$\min_{j\in\{1,2\}} e^{-\frac{1}{2\sigma_i^2}\|P_j-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|\left[0,P_{3_y}\right]-\mu_i\|^2}$
9	$e^{-\frac{1}{2\sigma_i^2}\|P_1-\mu_i\|^2}$	$e^{-\frac{1}{2\sigma_i^2}\|P_4-\mu_i\|^2}$