NON RIGID REGISTRATION OF SHAPES VIA DIFFEOMORPHIC POINT MATCHING AND CLUSTERING

Laurent Garcin
Institut Géographique National
Paris

Anand Rangarajan
Dept. of CISE
University of Florida

Laurent Younes
Center for Imaging Science
The Johns Hopkins University

ABSTRACT

We propose an algorithm achieving the matching of two point sets with unknown correspondence via the joint clustering of the two sets and estimation of a diffeomorphism linking the cluster centers via geodesic splines.

1. INTRODUCTION

Computing point correspondences is a recurrent problem in image processing, and has generated an important literature (eg. [1, 2, 3, 4, 5]). In [6], a new point of view on the correspondence problem is introduced, in which the quality of the matching is estimated by clustering the two sets of points and estimating a global deformation with spline matching techniques ([7]).

This method inherited the drawbacks of spline deformation methods which is to create ambiguities in the interpolated matching in presence of large deformations. Recent achievements of large-deformation landmark matching [8, 9, 10] provide mathematical tools and algorithms to estimate smooth and unambiguous landmark matching interpolations, even in the presence of very large deformations. In this paper, we provide a novel algorithm which combines the cluster-based approach of [6] and the geodesic splines derived in [9].

We will first model a matching energy (section 2) combining a clustering term (2.1) and a deformation term (2.2). We will then design an energy minimization algorithm (section 3) as an iteration of two steps (3.1 and 3.2) embedded in a deterministic annealing procedure. Results will be shown in section 4.

2. DEFINITION OF A MATCHING ENERGY

Let $X = (X_1, \ldots, X_K)$ and $Y = (Y_1, \ldots, Y_L)$ be two point sets to match in $\mathbb{R}^d$. We simultaneously cluster these point sets and, following the principles of large-deformation landmark matching, estimate an optimal set of trajectories between the centers of these clusters. More precisely, letting N be the number of clusters, we introduce N d-dimensional trajectories (or an Nd-dimensional trajectory) under the form $t \mapsto C(t) = (C_1(t), \ldots, C_N(t))$, for $t \in [0, 1]$, such that $C(0)$ provides the centers of the clusters in the first point set, and $C(1)$ those for the second point set (see fig. 1).

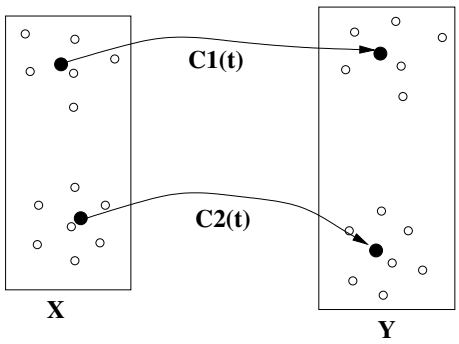


Fig. 1. Clusters trajectories

2.1. Data-Clusters Fitting Energy

Introduce M^X (resp. M^Y) the $K \times N$ (resp. $L \times N$) membership matrix of points of X (resp. Y) to clusters in $C(0)$ (resp. $C(1)$): M^X_{kn} is the degree of membership of point X_k in cluster n. Obviously we have the constraints : $\forall k, \sum_n M^X_{kn} = 1$ (resp. $\forall l, \sum_n M^Y_{ln} = 1$). The cluster energy is composed with two terms, the first measuring the cluster dispersion:

$$E_{clusters}(M^X, M^Y, C(0), C(1)) = \sum_{k,n} M^X_{kn} \|X_k - C_n(0)\|^2 + \sum_{l,n} M^Y_{ln} \|Y_l - C_n(1)\|^2$$

and the second is an entropy term controling the fuzziness of matrices M^X et M^Y :

$$E_{entropy}(M^X, M^Y) = \theta\big(\sum_{k,n} M^X_{kn} \log M^X_{kn} + \sum_{l,n} M^Y_{ln} \log M^Y_{ln}\big) \quad (1)$$

where θ is a temperature parameter. Matrices M^X and M^Y converge to binary matrices when the temperature parameter decreases to 0.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

2.2. Diffeomorphism Energy

This energy arises from the geodesic spline theory [9]. It involves a time dependent vector field $t \mapsto v(t,.)$ assuming to belong to a self reproducing kernel Hilbert space V, and is given by

$$E_{diff}(v,C) = \lambda \int_0^1 \|v(t,.)\|_V dt + \sum_n \int_0^1 \left\| \frac{dC_n}{dt} - v(t, C_n(t)) \right\|^2 dt$$

We denote G the reproducing kernel of V. In general, G is a matrix kernel but, for the sake of simplicity, we consider it as deriving from a scalar kernel i.e. $G = g.Id$, where g is a function from $\mathbb{R}^d \times \mathbb{R}^d$ to $\mathbb{R}$. The relation between G and the Hilbert space V is that, for all $v \in V$ (therefore taking values in $\mathbb{R}^d$) and for all $a \in \mathbb{R}^d$, for all $x \in \mathbb{R}^d$.

$$a^t v(x) = \langle v, G(.,x)a \rangle_V$$

with the implicit consequence that for all $a \in \mathbb{R}^d$, for all $x \in \mathbb{R}^d$, the vector field $g(.,x)a$ belongs to V. This yields a simpler form of the energy, depending only on the kernel g and on N finite dimensional trajectories $t \mapsto \alpha_n(t) \in \mathbb{R}^d$:

$$E_{diff}(\alpha, C) = \lambda \int_0^1 \sum_{n,m} g(C_n(t), C_m(t)) \langle \alpha_n(t), \alpha_m(t) \rangle dt$$
$$+ \sum_n \int_0^1 \left\| \frac{dC_n}{dt} - \sum_m g(C_n(t), C_m(t)) \alpha_m(t) \right\|^2 dt$$

We now discuss an energy minimization algorithm for $E_{clusters} + E_{entropy} + E_{diff}$. This will proceed by alternating a clustering step and a series of gradient descent steps for the deformation energy. For the last purpose, it is important to notice that, contrary to [9], the extremities of the trajectories C_n are not fixed.

2.3. Remarks

1. Using the fact that the minimized diffeomorphic energy can be seen as a geodesic distance in the set of landmark configurations, an equivalent formulation is to minimize

$$E_{clusters}(M^X, M^Y, C_X, C_Y) + E_{entropy}(M^X, M^Y) + d(C_X, C_Y)^2$$

where C_X and C_Y are the N cluster centers for X and Y and d is the geodesic distance on landmarks, defined by

$$d(C_X, C_Y)^2 = \min\{E_{diff}(v,C) : C(0) = C_X, C(1) = C_Y\}$$

2. A procedure with similar goals and a different approach has been proposed in [11]. This paper introduces intermediate energy terms to force the centers to be close to the landmark trajectories, wheras we propose here to fully integrate the geodesic deformation within the clustering process. This yields a formulation which is more compact, and a simpler algorithm.

3. ENERGY MINIMIZATION

3.1. Cluster Assignment Step

We proceed as in [6]. First compute the matrices $\hat{M}^X$ et $\hat{M}^Y$ minimizing the energy with fixed C and v :

$$\hat{M}^X_{kn} = \exp\left(-\frac{\|X_k - C_n(0)\|^2}{\theta}\right) \quad (2)$$

$$\hat{M}^Y_{ln} = \exp\left(-\frac{\|Y_l - C_n(1)\|^2}{\theta}\right) \quad (3)$$

Then normalize the rows of $\hat{M}^X$ and $\hat{M}^Y$ so that they sum up to 1. This yields the updated matrices M^X and M^Y.

3.2. Deformation Step

The deformation step can be divided into two steps.

3.2.1. Minimization in α

If we name $F(t)$ the matrix $(g(C_m(t), C_n(t)))_{mn}$ and :

$$A(t)^T = (\alpha_1(t), \ldots, \alpha_N(t))$$
$$C(t)^T = (C_1(t), \ldots, C_N(t))$$

the part of the energy involving the deformation (i.e. without the clusters part) becomes :

$$E(A) = \lambda \int_0^1 Tr(A(t)^T F(t) A(t)) + \int_0^1 \|C'(t) - F(t)A(t)\|^2 dt \quad (4)$$

and, similarly to [9], the minimization of the energy reduces to solving this linear system in A for all t :

$$\lambda A(t) + F(t)A(t) - C'(t) = 0 \quad (5)$$

3.2.2. Trajectory Step

We use a gradient descent to update the cluster trajectories. This requires computing the gradient of the energy E at C. To simplify the computation, we will let :

$$Z_m(t) = \frac{dC_m}{dt} - \sum_n F_{mn}(t)\alpha_n(t)$$
$$P_{mn}(t) = \nabla_1 g(C_m(t), C_n(t))$$

Let $H(\epsilon) = E(C + \epsilon D)$ and compute the derivative :

$$\begin{aligned}\frac{1}{2}\frac{dH}{d\epsilon} &= -\sum_{k,m} M^X_{km}\langle X_k - C_m(0), D_m(0)\rangle \\ &\quad - \sum_{l,m} M^Y_{lm}\langle Y_l - C_m(1), D_m(1)\rangle \\ &\quad + \lambda \int_0^1 \sum_{m,n}\langle P_{mn}(t), D_m(t)\rangle\langle \alpha_m(t), \alpha_n(t)\rangle dt \\ &\quad - \int_0^1 \sum_{m,n}\langle P_{mn}(t), D_m(t)\rangle\langle Z_m(t), \alpha_n(t)\rangle dt \\ &\quad - \int_0^1 \sum_{m,n}\langle P_{mn}(t), D_n(t)\rangle\langle Z_m(t), \alpha_n(t)\rangle dt \\ &\quad + \int_0^1 \sum_m \left\langle Z_m(t), \frac{dD_m}{dt}\right\rangle dt\end{aligned} \tag{6}$$

We assume that the gradient step is performed immediately after the deformation step so that : $\forall m, Z_m = \lambda\alpha_m$ (cf. eqn. (5)). If we also use the symmetry of the kernel G and if we integrate the last term by part, the expression of the derivative becomes :

$$\begin{aligned}\frac{1}{2}\frac{dH}{d\epsilon} &= -\sum_{k,m} M^X_{km}\langle X_k - C_m(0), D_m(0)\rangle \\ &\quad - \sum_{l,m} M^Y_{lm}\langle Y_l - C_m(1), D_m(1)\rangle \\ &\quad - \lambda \int_0^1 \sum_{m,n}\langle P_{mn}(t), D_m(t)\rangle\langle \alpha_m(t), \alpha_n(t)\rangle dt \\ &\quad - \lambda \int_0^1 \sum_m \left\langle \frac{d\alpha_m}{dt}, D_m(t)\right\rangle dt \\ &\quad + \lambda \sum_m \langle \alpha_m(1), D_m(1)\rangle - \langle \alpha_m(0) D_m(0)\rangle\end{aligned} \tag{7}$$

One deduces the gradient from the last expression :

$$\begin{aligned}\frac{1}{2}(\nabla_C E)_m(t) &= -\lambda \sum_n P_{mn}(t)\langle \alpha_m(t), \alpha_n(t)\rangle - \lambda\frac{d\alpha_m}{dt} \\ &\quad - \left(\sum_k M^X_{km}(X_k - C_m(0)) + \lambda\alpha_m(0)\right)\delta_0(t) \\ &\quad - \left(\sum_l M^Y_{lm}(Y_l - C_m(1)) - \lambda\alpha_m(1)\right)\delta_1(t)\end{aligned} \tag{8}$$

where δ_0 et δ_1 are Dirac distributions at 0 and 1.

3.3. Algorithm

Here is the algorithmic scheme :

Compute the spline matrices $F(t)$
Initialize vectors $\alpha_n(t)$
repeat
  repeat
    Update deformation (α) by least squares algorithm
    Update cluster trajectories C_n with one gradient step
    Update spline matrices $F(t)$
  until $\Delta E < \epsilon$
  Update cluster membership matrices M^X and M^Y
  Decrease temperature θ by factor θ_d
until $\theta < \theta_f$

4. RESULTS

In our applications, we use a time discretization of $T = 10$ steps and a gaussian kernel. We first show the superiority of the geodesic splines with respect to the classical splines (which correspond to $T = 1$ time step). The input data are vectorizations fig. 2(a) and 2(b) and the retrieved deformations are shown in fig. 2(c) and 2(d). One can note that the diffeomorphic constraint yields the expected deformation in fig. 2(c) whereas the deformation in fig. 2(d) is clearly not a diffeomorphism. The final energy ($E = 0,226124$) in the geodesic case is less than the final energy in the classical case ($E = 0,450852$).

We also test our algorithm on vectorizations of fig. 3(a) and 3(b) ($K = 521, L = 474, N = 100$) The clusters at $t = 0$ are shown in fig. 3(c) and 3(e) and the clusters at $t = 1$ and corresponding deformation in fig. 3(d) and 3(f).

5. REFERENCES

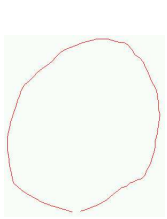
[1] G Shapiro, L and M Haralik, R, "Structural description and inexact matching," *IEEE TPAMI*, vol. 3, no. 4, pp. 504–519, 1981.

[2] C von der Malsburg, "Pattern recognition by labeled graph matching," *Neural Networks*, vol. 1, pp. 141–148, 1988.

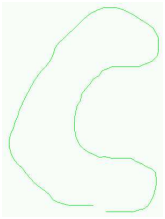
[3] S Baird, H, *Model-Based Image Matching Using Location*, MIT Press, 1985.

[4] B Bhanu and O Faugeras, "Shape matching of two-dimensional objects," *IEEE TPAMI*, vol. 6, pp. 137–156, 1984.

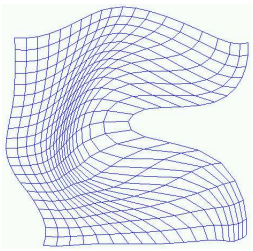
[5] S Davis, L, "Shape matching using relaxation techniques," *IEEE TPAMI*, vol. 1, no. 1, pp. 60–72, 1979.



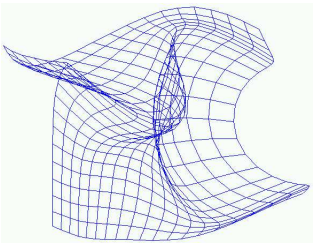
(a) Set X



(b) Set Y



(c) Geodesic splines



(d) Classical splines

Fig. 2. Geodesic splines vs. classical splines

[6] Haili Chui, Larry Win, Robert Schultz, Jim Duncan, and Anand Rangarajan, "A unified non-rigid feature registration method for brain mapping," *Medical Image Analysis*, 2002.

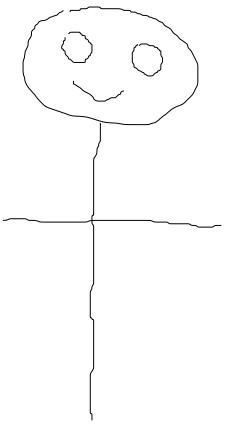
[7] L. Bookstein, *Morphometric tools for landmark data, geometry and biology*, Cambridge University Press, 1991.

[8] S Joshi and M Miller, "Landmark matching via large deformation diffeomorphisms," *IEEE transactions in image processing*, vol. 9, no. 8, pp. 1357–1370, 2000.

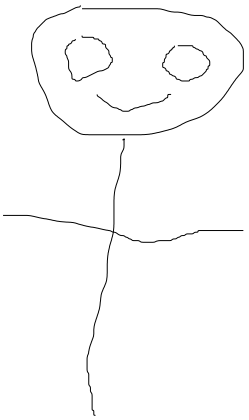
[9] Laurent Younes and Vincent Camion, "Geodesic interpolating splines," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. CVPR, 2001, pp. 513–527, Springer.

[10] M. F. Beg, M. Miller, A. Trouvé, and L. Younes, "The Euler–Lagrange equations for interpolating sequence of landmark datasets," in *MICCAI 2003*, 2003.

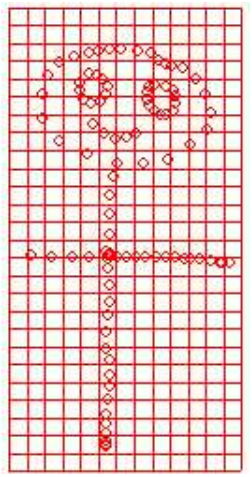
[11] H Guo, A Rangarajan, S Joshi, and L Younes, "Non-rigid registration of shapes via diffeomorphic point-matching," in *ISBI 04*, 2004.



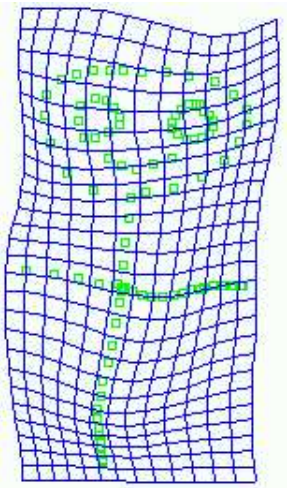
(a) Set X



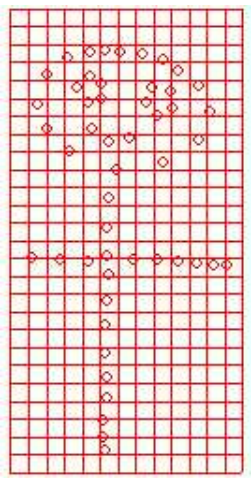
(b) Set Y



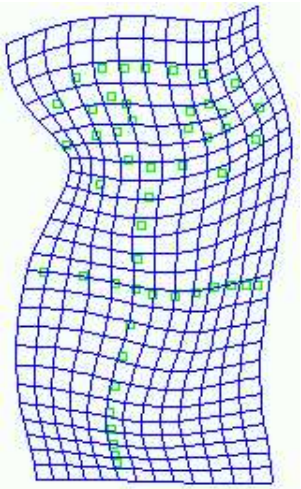
(c) Cluter centers $C(0)$ with $N = 100$



(d) Cluter centers $C(1)$ with $N = 100$



(e) Cluter centers $C(0)$ with $N = 50$



(f) Cluter centers $C(1)$ with $N = 50$

Fig. 3.