

A modified ICP algorithm for normal-guided surface registration

Daniel Münch, Benoît Combès, Sylvain Prima

► **To cite this version:**

Daniel Münch, Benoît Combès, Sylvain Prima. A modified ICP algorithm for normal-guided surface registration. Medical Imaging 2010: Image Processing, Feb 2010, San Diego, United States. pp.76231A, 2010, <10.1117/12.844994>. <inserm-00589939>

HAL Id: inserm-00589939

<http://www.hal.inserm.fr/inserm-00589939>

Submitted on 2 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Modified ICP Algorithm for Normal-Guided Surface Registration

Daniel Münch, Benoît Combès and Sylvain Prima
INSERM, U746, F-35042 Rennes, France
INRIA, VisAGeS Project-Team, F-35042 Rennes, France
University of Rennes I, CNRS, UMR 6074, IRISA, F-35042 Rennes, France
[bcombes,sprima]@irisa.fr, <http://www.irisa.fr/visages>

ABSTRACT

The iterative closest point (ICP) algorithm is probably the most popular algorithm for fine registration of surfaces. Among its key properties are: a simple minimization scheme, proofs of convergence as well as the easiness to modify and improve it in many ways (*e.g.* use of fuzzy point correspondences, incorporation of a priori knowledge, extensions to non-linear deformations, speed-up strategies, *etc.*) while keeping the desirable properties of the original method. However, most ICP-like registration methods suffer from the fact that they only consider the distance between the surfaces to register in the criterion to minimize, and thus are highly dependent on how the surfaces are aligned in the first place. This explains why these methods are likely to be trapped in local minima and to lead to erroneous solutions. A solution to partly alleviate this problem would consist in adding higher-order information in the criterion to minimize (*e.g.* normals, curvatures, *etc.*), but previous works along these research tracks have led to computationally intractable minimization schemes. In this paper, we propose a new way to include the point unit normals in addition to the point coordinates to derive an ICP-like scheme for non-linear registration of surfaces, and we show how to keep the properties of the original ICP algorithm. Our algorithm rests on a simple formula showing how the unit normal changes when a surface undergoes a small deformation. The use of this formula in an ICP-like algorithm is made possible by adequate implementation choices, most notably the use of a local, differentiable, parametrization of the surfaces and a locally affine deformation model using this local parametrization. Then we experimentally show the strong added value of using the unit normals in a series of controlled experiments.

1. INTRODUCTION

Non-linear registration is a key tool for many applications in medical imaging: morphometric analyses, follow-up studies, computer-assisted surgery, *etc.* Surfaces represented by point clouds or meshes are especially convenient to represent a subject's anatomy, as they allow to focus the subsequent analyses on the structure of interest. Many surface registration algorithms have been proposed in the literature. One of the most popular is the ICP algorithm proposed by Besl & McKay.¹ Over the years it has led to numerous successful variants and extensions providing many ICP-like algorithms using fuzzy correspondences,² a priori knowledge,³ selection of points for faster convergence,⁴ robust techniques to deal with outliers,^{5,6} *etc.* Extensions to non-linear deformations have also been successfully developed.^{2,3,7} In essence, the ICP algorithm minimizes the distance between the two surfaces. However, some works have tried to add higher-order information in the ICP criterion, to alleviate the well-known local convergence problem of the algorithm, but they have led to intractable solutions.

For instance, Feldmar & Ayache have shown how to include the normals and curvatures.⁸ After showing how these quantities are modified by a locally affine deformation, they devised a criterion combining point coordinates, normals and curvatures and showed how to minimize it in an iterative manner similarly to the original ICP algorithm. However, solving for the optimal transformation given correspondences in this framework is a non-linear problem which makes the overall scheme computationally expensive and difficult to implement in case of large surfaces. Another significant work along these lines has been done by Kambhamettu and colleagues in a series of papers⁹⁻¹⁶ where they showed how to properly derive a simple solution to this last problem, using linear algebra, when the hypothesis of a special type of deformations (a subclass of affine transformations) is made. However, in their work this solution was not encompassed in an ICP-type algorithm, and an exhaustive search over all possible pairs of correspondences was made instead to register the two surfaces. Moreover, the

point-to-point distances was not taken into account in their algorithm, while these are of the utmost importance when the normals cannot be faithfully estimated.

In this paper, 1) we propose to use the locally affine transformations subtype proposed by Kambhamettu and colleagues to devise a proper ICP-type registration criterion using these transformations, the point coordinates and the point unit normals, 2) we show how to derive an iterative solution to optimise this criterion, possessing the desirable convergence and simplicity properties of the original ICP algorithm, and 3) we experimentally show that adding the unit normals in the ICP criterion significantly improves the registration.

2. GENERAL SCHEME

The problem of surface registration consists in finding the transformation which best superposes a surface P with a reference surface X . In this paper we consider surfaces represented by point sets or triangle meshes. We call these sets $P = \{\mathbf{p}_i\}, i = 1 \dots N_P$ and $X = \{\mathbf{x}_j\}, j = 1 \dots N_X$. A very fruitful approach to tackle this problem is to introduce point correspondences as intermediate variables. Following this idea, Besl & McKay have proposed the ICP algorithm, whose criterion to minimize has been later modified by Chui and Rangarajan² as well as Combès and Prima,³ and that can be formulated in a very general way as follows:

$$\varepsilon(T, \mathbf{A}) = \sum_{i=1}^{N_P} \sum_{j=1}^{N_X} A_{ij} d_T(\mathbf{x}_j, \mathbf{p}_i) + \beta L(T) + \gamma \sum_{i=1}^{N_P} \sum_{j=1}^{N_X} A_{ij} \log(A_{ij}) \quad \text{with } \forall i, \sum_j A_{ij} = 1 \text{ and } \forall(i, j), A_{ij} \geq 0 \quad (1)$$

- T is the transformation best superposing the surfaces (and that can be rigid, affine, elastic, *etc.*).
- \mathbf{A} is a $N_P \times N_X$ matrix that describes the *fuzzy* correspondences between the surface points. The greater A_{ij} , the more likely the point $\mathbf{x}_j \in X$ to be the correspondent of the point $\mathbf{p}_i \in P$. \mathbf{A} can be viewed as a hidden variable of the registration problem and will be called the match matrix.
- $d_T(\mathbf{x}_j, \mathbf{p}_i)$ is a general distance measuring the *discrepancy* between \mathbf{x}_j and $T(\mathbf{p}_i)$.
- L is a regularizer allowing to penalize discontinuities of T over the space, with $\beta \geq 0$ weighing its influence.
- $\sum_{i=1}^{N_P} \sum_{j=1}^{N_X} A_{ij} \log(A_{ij})$ is a barrier function allowing to control the *fuzziness* of \mathbf{A} (the higher $\gamma \geq 0$, the greater the fuzziness). In practice, this term convexifies the criterion.

The strategy first proposed by Besl & McKay¹ to minimize ε consists in alternating the estimation of the point-to-point fuzzy correspondences \mathbf{A} and of the transformation T . This has been shown to converge to an at least local minimum of the criterion ε . The iterative algorithm can then be simply written as follows:

- | |
|---|
| <p>Step 0: Initialize T^0</p> <p>Step 1: $\mathbf{A}^{n+1} = \arg \min_{\mathbf{A}} \varepsilon(T^n, \mathbf{A})$</p> <p>Step 2: $T^{n+1} = \arg \min_T \varepsilon(T, \mathbf{A}^{n+1})$</p> <p>Step 3: if $T^{n+1} - T^n \geq \mu$ go to Step 1 else exit</p> |
|---|

Solving Steps 1 and 2 mainly depends on the choice of d_T , L and T , and the main challenge in this framework is to make realistic choices while keeping Step 1 and Step 2 computationally tractable. In the original ICP algorithm, d_T is simply the Euclidean distance between \mathbf{x}_j and $T(\mathbf{p}_i)$, T is a rigid-body transformation, L is the null function and γ is set to zero. Step 1 is then solved setting A_{ij}^{n+1} to 1 if and only \mathbf{x}_j is the closest point of $T^n(\mathbf{p}_i)$ in X and zero else (typically using a kd-tree). Solving Step 2 then consists in finding the rigid-body transformation best superposing the correspondences established during Step 1 in the least-squares sense, which can be done in a closed-form using for instance unit quaternions^{17,18} or the SVD.¹⁹

This iterative algorithm is very satisfying, as it is very simple and can be easily extended to cope with some specific types of non-linear transformations. However, the lack of a good initial transformation T makes it likely to be trapped in a local minimum, mostly because d_T is only based on the Euclidean distance between point coordinates. In the following we propose to include the unit normals in addition to the point coordinates to devise a new distance term d_T . We show how an adequate transformation model and regularization term can be used to keep Steps 1 and 2 of the above-mentioned iterative scheme tractable.

3. CHOICE OF THE DISTANCE TERM

Instead of using the Euclidean distance between surface points $d_T(\mathbf{x}_j, \mathbf{p}_i) = \|\mathbf{x}_j - T(\mathbf{p}_i)\|^2$ (as in the original ICP), we propose to use instead: $d_T(\mathbf{x}_j, \mathbf{p}_i) = \|\mathbf{x}_j - T(\mathbf{p}_i)\|^2 + \lambda \|\mathbf{n}_{\mathbf{x}_j} - \mathbf{n}_{T(\mathbf{p}_i)}\|^2$ where $\mathbf{n}_{\mathbf{x}_j}$ and $\mathbf{n}_{T(\mathbf{p}_i)}$ are the unit normal vectors at points $\mathbf{x}_j \in X$ and $T(\mathbf{p}_i) \in T(P)$, and λ is a positive value weighing the influence of the normal-to-normal distance term over the point-to-point distance term.

It turns out that $\mathbf{n}_{T(\mathbf{p}_i)}$ can be expressed simply in terms of $\mathbf{n}_{\mathbf{p}_i}$ (unit normal vector at point $\mathbf{p}_i \in P$) if additional hypotheses are made. We rely on the work by Kambhamettu and colleagues⁹⁻¹⁶ to derive this expression. First we assume that:

- The surface P is given in parametric form, and each point \mathbf{p} is a vector-valued differentiable function of parameters u and v .
- This parametrization is orthogonal, *i.e.* $\mathbf{p}_u \cdot \mathbf{p}_v = 0$ (\mathbf{p}_u and \mathbf{p}_v are the derivatives of \mathbf{p} with respect to u and v respectively and \cdot is the dot product).
- The position of point \mathbf{p} after the transformation is $T(\mathbf{p}) = \mathbf{p} + \mathbf{s}$, where \mathbf{s} is a displacement function capturing the local deformation of the surface P .
- The deformation of P is *small*, *i.e.* all second-order quantities involving \mathbf{s} in the computations are neglected.

Then $\mathbf{n}_{T(\mathbf{p}_i)}$ can be shown to be equal to:

$$\mathbf{n}_{T(\mathbf{p}_i)} = \mathbf{n}_{\mathbf{p}_i} - \mathbf{n}_{\mathbf{p}_i} \times \left(\frac{\mathbf{p}_{u_i} \times \mathbf{s}_{u_i}}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} + \frac{\mathbf{p}_{v_i} \times \mathbf{s}_{v_i}}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \right) \quad (2)$$

In the notation, \times is the cross product, \cdot the dot product, and u_i and v_i are the parameters at point \mathbf{p}_i . Note that this formula is independent of the chosen parametrization, provided it is orthogonal. The orthogonality of the parametrization and the hypothesis of a small deformation mostly allow to simplify the computations. Relaxing these constraints would be possible, at the cost of more complex computations.²⁰ We explain in Section 5 how to derive a local differentiable orthogonal parametrization for P .

As a summary, we design the distance term d_T as follows:

$$d_T(\mathbf{x}_j, \mathbf{p}_i) = \|\mathbf{x}_j - \mathbf{p}_i - \mathbf{s}_i\|^2 + \lambda \left\| \mathbf{n}_{\mathbf{x}_j} - \mathbf{n}_{\mathbf{p}_i} - \mathbf{n}_{\mathbf{p}_i} \times \left(\frac{\mathbf{p}_{u_i} \times \mathbf{s}_{u_i}}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} + \frac{\mathbf{p}_{v_i} \times \mathbf{s}_{v_i}}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \right) \right\|^2 \quad (3)$$

4. MINIMIZATION OF THE CRITERION

4.1 Derivatives of the Distance and Regularization Terms w.r.t. the Transformation

Before tackling the minimization of the whole ICP criterion, we need to further specify the deformation model and the associated regularization term. As suggested by Kambhamettu and colleagues,⁹⁻¹⁶ we propose to use a locally affine deformation model:

$$\mathbf{s}(u_i, v_i) = u_i \mathbf{a}_i + v_i \mathbf{b}_i$$

We also propose to design a first-order regularization term that we define as:

$$L(T) = \sum_{(k,l) \in C_2(P)} \|\mathbf{u}_k \mathbf{a}_k + v_k \mathbf{b}_k - (\mathbf{u}_l \mathbf{a}_l + v_l \mathbf{b}_l)\|^2$$

where $C_2(P)$ is the set of second-order cliques of P . In practice, the cliques are determined using a mesh: $(k, l) \in C_2 \Leftrightarrow$ there exists an edge between points \mathbf{p}_k and \mathbf{p}_l . These choices are especially convenient, as they allow Step 2 of our ICP-type algorithm to stay tractable, as we will see in Section 4.2.2.

Below, we provide the derivatives of the distance and regularization terms with respect to the transformation parameters \mathbf{a}_i and \mathbf{b}_i . Using \mathbf{I} as the identity matrix in $\mathbb{R}^{3 \times 3}$ and writing everything in block matrix form, we get:

- Derivatives of the point coordinates term:

$$\left(\begin{array}{c} \frac{\partial}{\partial \mathbf{a}_i} \\ \frac{\partial}{\partial \mathbf{b}_i} \end{array} \right) \|\mathbf{x}_j - \mathbf{p}_i - (u_i \mathbf{a}_i + v_i \mathbf{b}_i)\|^2 =$$

$$\left(\begin{array}{c} -2u_i[\mathbf{x}_j - \mathbf{p}_i - (u_i \mathbf{a}_i + v_i \mathbf{b}_i)] \\ -2v_i[\mathbf{x}_j - \mathbf{p}_i - (u_i \mathbf{a}_i + v_i \mathbf{b}_i)] \end{array} \right) = 2 \left(\begin{array}{cc} u_i^2 \mathbf{I} & u_i v_i \mathbf{I} \\ u_i v_i \mathbf{I} & v_i^2 \mathbf{I} \end{array} \right) \left(\begin{array}{c} \mathbf{a}_i \\ \mathbf{b}_i \end{array} \right) - 2 \left(\begin{array}{c} u_i[\mathbf{x}_j - \mathbf{p}_i] \\ v_i[\mathbf{x}_j - \mathbf{p}_i] \end{array} \right) \quad (4)$$

- Derivatives of the point unit normal term:

$$\left(\begin{array}{c} \frac{\partial}{\partial \mathbf{a}_i} \\ \frac{\partial}{\partial \mathbf{b}_i} \end{array} \right) \left\| \mathbf{n}_{\mathbf{x}_j} - \mathbf{n}_{\mathbf{p}_i} - \mathbf{n}_{\mathbf{p}_i} \times \left(\frac{\mathbf{p}_{u_i} \times \mathbf{s}_{u_i}}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} + \frac{\mathbf{p}_{v_i} \times \mathbf{s}_{v_i}}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \right) \right\|^2 =$$

$$2 \left(\begin{array}{cc} \frac{\mathbf{n}_{\mathbf{p}_i} \mathbf{n}_{\mathbf{p}_i}^T}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} & \mathbf{0} \\ \mathbf{0} & \frac{\mathbf{n}_{\mathbf{p}_i} \mathbf{n}_{\mathbf{p}_i}^T}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \end{array} \right) \left(\begin{array}{c} \mathbf{a}_i \\ \mathbf{b}_i \end{array} \right) + 2 \left(\begin{array}{c} \frac{(\mathbf{n}_{\mathbf{x}_j} - \mathbf{n}_{\mathbf{p}_i}) \cdot \mathbf{p}_{u_i}}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} \mathbf{n}_{\mathbf{p}_i} \\ \frac{(\mathbf{n}_{\mathbf{x}_j} - \mathbf{n}_{\mathbf{p}_i}) \cdot \mathbf{p}_{v_i}}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \mathbf{n}_{\mathbf{p}_i} \end{array} \right) \quad (5)$$

- Derivatives of the regularization term:

$$\left(\begin{array}{c} \frac{\partial}{\partial \mathbf{a}_i} \\ \frac{\partial}{\partial \mathbf{b}_i} \end{array} \right) \sum_{(k,l) \in C_2(P)} \|u_k \mathbf{a}_k + v_k \mathbf{b}_k - (u_l \mathbf{a}_l + v_l \mathbf{b}_l)\|^2 =$$

$$2|V(i)| \left(\begin{array}{cc} u_i^2 \mathbf{I} & u_i v_i \mathbf{I} \\ u_i v_i \mathbf{I} & v_i^2 \mathbf{I} \end{array} \right) \left(\begin{array}{c} \mathbf{a}_i \\ \mathbf{b}_i \end{array} \right) - 2 \left(\begin{array}{c} u_i \sum_{k \in V(i)} (u_k \mathbf{a}_k + v_k \mathbf{b}_k) \\ v_i \sum_{k \in V(i)} (u_k \mathbf{a}_k + v_k \mathbf{b}_k) \end{array} \right) \quad (6)$$

where $V(i)$ contains the indices of neighbors of point \mathbf{p}_i in P and $|V(i)|$ is its cardinality.

4.2 Minimization of the Criterion

Using the choices made in Sections 3 and 4.1 for the distance and the regularization terms, the general criterion (1) becomes:

$$\varepsilon(T, \mathbf{A}) = \sum_{i=1}^{N_P} \sum_{j=1}^{N_X} A_{ij} \left(\|\mathbf{x}_j - \mathbf{p}_i - (u_i \mathbf{a}_i + v_i \mathbf{b}_i)\|^2 + \lambda \left\| \mathbf{n}_{\mathbf{x}_j} - \mathbf{n}_{\mathbf{p}_i} - \mathbf{n}_{\mathbf{p}_i} \times \left(\frac{\mathbf{p}_{u_i} \times \mathbf{s}_{u_i}}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} + \frac{\mathbf{p}_{v_i} \times \mathbf{s}_{v_i}}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \right) \right\|^2 \right)$$

$$+ \beta \sum_{(k,l) \in C_2(P)} \|u_k \mathbf{a}_k + v_k \mathbf{b}_k - (u_l \mathbf{a}_l + v_l \mathbf{b}_l)\|^2 + \gamma \sum_{i=1}^{N_P} \sum_{j=1}^{N_X} A_{ij} \log(A_{ij}) \quad (7)$$

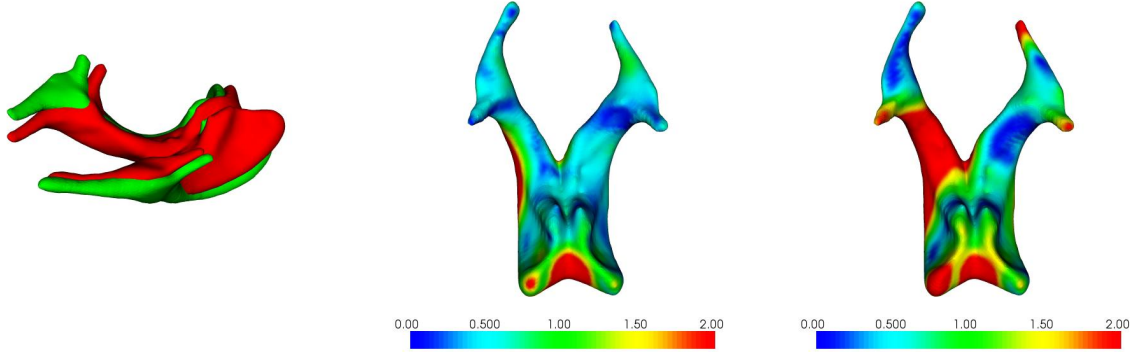
We now proceed to the actual iterative minimization of this criterion, by first computing the optimal match matrix \mathbf{A} when the transformation is given, and second computing the optimal transformation T for a given match matrix. These optima are computed as values where the derivatives vanish.

4.2.1 Step 1: Point Correspondences

Minimizing the criterion with respect to A_{ij} given the constraints $\forall i, \sum_j A_{ij} = 1$ and $\forall(i, j), A_{ij} \geq 0$ can be done in a straightforward way using Lagrange multipliers and give the following results:

$$\forall(i, j), \frac{A_{ij}}{\alpha_j} = \exp \left(\frac{\|\mathbf{x}_j - \mathbf{p}_i - (u_i \mathbf{a}_i + v_i \mathbf{b}_i)\|^2 + \lambda \left\| \mathbf{n}_{\mathbf{x}_j} - \mathbf{n}_{\mathbf{p}_i} - \mathbf{n}_{\mathbf{p}_i} \times \left(\frac{\mathbf{p}_{u_i} \times \mathbf{s}_{u_i}}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} + \frac{\mathbf{p}_{v_i} \times \mathbf{s}_{v_i}}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \right) \right\|^2}{-\gamma} \right) \quad (8)$$

where α_j is such that $\sum_j A_{ij} = 1$. To reduce the computational cost of estimating the optimal transformation in Step 2, we choose to compute one-to-one correspondences from the fuzzy match matrix \mathbf{A} , as proposed by Chui & Rangarajan.² That is once we have estimated \mathbf{A} we consider that \mathbf{p}_i is matched with the *virtual* point $\hat{\mathbf{x}}_i = \sum_j A_{ij} \mathbf{x}_j$. Thus we consider that one-to-one correspondences between P and S are achieved in this first step.



(a) Original (green) and deformed (red) ventricles (b) Registration with our algorithm after 10 iterations (c) Registration with the reference algorithm after 120 iterations

Figure 1. Mapping of registration errors for rotated (4 degrees) and non-linearly deformed (TPS) ventricles.

4.2.2 Step 2: Transformation

After matching each point $\mathbf{p}_i \in P$ with a point $\hat{\mathbf{x}}_i$, we use the formulas (4), (5) and (6) obtained in Section 4.1 to minimize the criterion (7) with respect to \mathbf{a}_i and \mathbf{b}_i , $\forall i \in 1 \dots N_P$. This yields a system of $6N_P$ equations:

$$\forall i \in 1 \dots N_P, \begin{pmatrix} (1 + \beta|V(i)|)u_i^2\mathbf{I} + \lambda \frac{\mathbf{n}_{\mathbf{p}_i} \mathbf{n}_{\mathbf{p}_i}^T}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} & (1 + \beta|V(i)|)u_i v_i \mathbf{I} \\ (1 + \beta|V(i)|)u_i v_i \mathbf{I} & (1 + \beta|V(i)|)v_i^2\mathbf{I} + \lambda \frac{\mathbf{n}_{\mathbf{p}_i} \mathbf{n}_{\mathbf{p}_i}^T}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \end{pmatrix} \begin{pmatrix} \mathbf{a}_i \\ \mathbf{b}_i \end{pmatrix} = \begin{pmatrix} u_i[(\hat{\mathbf{x}}_i - \mathbf{p}_i) + \beta \sum_{k \in V(i)} (u_k \mathbf{a}_k + v_k \mathbf{b}_k)] - \lambda \frac{(\mathbf{n}_{\hat{\mathbf{x}}_i} - \mathbf{n}_{\mathbf{p}_i}) \cdot \mathbf{p}_{u_i}}{\mathbf{p}_{u_i} \cdot \mathbf{p}_{u_i}} \mathbf{n}_{\mathbf{p}_i} \\ v_i[(\hat{\mathbf{x}}_i - \mathbf{p}_i) + \beta \sum_{k \in V(i)} (u_k \mathbf{a}_k + v_k \mathbf{b}_k)] - \lambda \frac{(\mathbf{n}_{\hat{\mathbf{x}}_i} - \mathbf{n}_{\mathbf{p}_i}) \cdot \mathbf{p}_{v_i}}{\mathbf{p}_{v_i} \cdot \mathbf{p}_{v_i}} \mathbf{n}_{\mathbf{p}_i} \end{pmatrix} \quad (9)$$

The point $\hat{\mathbf{x}}_i$ does not necessarily belong to the surface X . To compute $\mathbf{n}_{\hat{\mathbf{x}}_i}$, we simply project the point $\hat{\mathbf{x}}_i$ on the surface X and use the unit normal at this point in the formulas. The complete set of formulas obtained when keeping the entire fuzzy match matrix as computed in Step 1 using Equation (8) would be easy to derive, but we omit them for the sake of simplicity.

It could appear at first that the 6 equations obtained by differentiating with respect to the unknowns \mathbf{a}_i and \mathbf{b}_i are enough to compute them. It is not the case, because \mathbf{a}_i and \mathbf{b}_i also appear in the equations obtained when differentiating with respect to \mathbf{a}_k and \mathbf{b}_k , when $k \in V(i)$. Thus, in order to minimize the criterion with respect to the set $(\mathbf{a}_i, \mathbf{b}_i)$, $i \in 1 \dots N_P$ we use an Iterated Coordinate Descent (ICD) algorithm, where we minimize with respect to each pair of parameters $(\mathbf{a}_i, \mathbf{b}_i)$ in turn, while keeping all the other pairs fixed:

- Step 2.0:** Initialize $(\mathbf{a}_i, \mathbf{b}_i) \forall i \in 1 \dots N_P$
- Step 2.1:** $\forall i \in 1 \dots N_P$ $\tilde{\mathbf{a}}_j, \tilde{\mathbf{b}}_j = \arg \min_{\mathbf{a}_i, \mathbf{b}_i} \varepsilon((\tilde{\mathbf{a}}_0, \tilde{\mathbf{b}}_0) \dots (\mathbf{a}_i, \mathbf{b}_i) \dots (\tilde{\mathbf{a}}_{N_P}, \tilde{\mathbf{b}}_{N_P}), \mathbf{A})$
- Step 2.2:** If $|\tilde{\varepsilon} - \varepsilon| \geq \mu$ go to Step 2.0 else exit

Thus, each Step 2.1 consists of N_P linear systems with a simple 6×6 matrix inversion, given by Equation (9). It can be shown that using the ICD algorithm does not change the convergence properties of the algorithm.

5. IMPLEMENTATION DETAILS

We look for a Monge patch representation at each point \mathbf{p} of the surface P . Such a representation is of the form $(x, y, z) = (u, v, f(u, v))$ and is an especially simple type of orthogonal parametrization. To obtain this representation, we first compute the unit normal at each point $\mathbf{p} \in P$ using the widely used VTK library

Rotation in degrees	Rotation + TPS deformation					
	4	5	6	7	8	9
Our algorithm (10 iterations)	0.52	0.56	0.63	0.72	0.83	0.98
Reference algorithm (10 iterations)	1.63	1.87	2.35	3.67	5.48	6.40
Reference algorithm (120 iterations)	1.04	1.10	1.29	1.64	4.32	5.48

Table 1. Mean error of registration for rotated, non-linearly deformed (using thin plate splines) ventricles using our algorithm (after 10 iterations) and using the reference algorithm (after 10 and 120 iterations).

(<http://www.vtk.org>); the same is done for each point $\mathbf{x} \in X$. Then, using the unit normal at point \mathbf{p} , we define a local coordinate system (where \mathbf{p} is at $(0, 0, 0)$ and the unit normal lies along the positive z -axis) and we fit a quadratic surface of the type $f(u, v) = Au^2 + Buv + Cv^2$ in the least-squares sense using the neighbors of \mathbf{p} .²¹

We use the point coordinates and the unit normals of surface X to build a kd-tree ($k = 6$) for an efficient implementation of Equation (8). The use of a kd-tree is made possible by introducing a threshold function

$$\rho_\delta(x) = \begin{cases} x & : |x| < \delta \\ \delta & : \text{else} \end{cases} \quad \text{and thus replacing the original distance term by a term of the form } d_{T,\delta}(\cdot) = \rho_\delta(d_T(\cdot)).$$

It can be shown that this change does not affect the convergence properties of the algorithm.²² It allows us to use range queries in the kd-tree to compute the values A_{ij} and thus reduce the time complexity of the algorithm. Moreover, the matrix \mathbf{A} becomes sparse, reducing the space complexity. Another positive side effect of this threshold function is to make the algorithm less vulnerable to outliers. Note that when γ becomes close to 0, Equation (8) boils down to a simple nearest neighbor query.

6. EXPERIMENTS AND RESULTS

As outlined in the introduction, the convergence of the classical ICP algorithm depends on the initial position of the surfaces: badly aligned surfaces in the first place may trap the algorithm in a local minimum. We show in this section that our algorithm may help to avoid this kind of problems. We generate the data in the following way:

- We segment several lateral ventricles from 3T T1-weighted brain isotropic MRI data using itk-SNAP (<http://www.itksnap.org>), which gives surfaces X of about 10,000 points.
- We generate P from X by applying 1) a random thin plate spline transformation (to simulate such a transformation, we select a set of 8 landmarks on X and randomly move them independently in a sphere of radius 20 mm around their initial position) and then 2) rotations with increasing angles. This gives us ground truth pairs (X, P) .
- We register each such obtained pair using 1) our algorithm (with the following parameters: $\beta = 50$, $\gamma = 0$, $\lambda = 1000$) and 2) a *reference* algorithm (where $\lambda = 0$, *i.e.* only the point coordinates are used to register X and P).
- We evaluate the registration error by computing the mean distance between homologous points after registration. The values shown in Table 1 are the mean of these mean errors over the different ventricle pairs and results on one of the pairs are displayed on Figure 1.

The conclusions of these preliminary experiments are twofold: 1) the accuracy of our algorithm is much higher and 2) its convergence is much faster compared to the reference algorithm.

7. DISCUSSION

In this paper we have presented a novel algorithm for non-linear registration of surfaces. We have shown how to include the unit normals in addition to the point coordinates in an ICP-type criterion and we have shown how to minimize it in a simple way when a special type of small deformations is assumed between the two surfaces. Numerous adaptations and variants proposed in the literature for the original ICP algorithm could be applied in a straightforward way to our algorithm, such as the use of a prior affinity distance term as proposed by Combès

& Prima.³ Future work could also consist in adding a term based on the curvature in the criterion, using another set of formulas derived by Kambhamettu and colleagues. We have presented a set of preliminary experiments, and extending the validation dataset would be necessary. Evaluating different sets of parameters and different multiresolution or multiscale schemes would also be needed. At last, our algorithm needs to be compared with others, especially with landmark-free approaches²³ and with other methods that do not resort to point-to-point correspondences.²⁴

REFERENCES

- [1] Besl, P. and McKay, H., "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**(2), 239–256 (1992).
- [2] Chui, H. and Rangarajan, A., "A new point matching algorithm for non-rigid registration," *Computer Vision and Image Understanding* **89**(2-3), 114–141 (2003).
- [3] Combès, B. and Prima, S., "Prior affinity measures on matches for ICP-like nonlinear registration of free-form surfaces," in [*IEEE International Symposium on Biomedical Imaging*], 370–373 (June 2009).
- [4] Rusinkiewicz, S. and Levoy, M., "Efficient Variants of the ICP Algorithm," in [*International Conference on 3D Digital Imaging and Modeling*], 145–152 (May 2001).
- [5] Chetverikov, D., Svirko, D., Stepanov, D., and Krsek, P., "The trimmed iterative closest point algorithm," in [*IEEE International Conference on Pattern Recognition*], Kasturi, R., Laurendeau, D., and Suen, C., eds., 545–548 (Aug. 2002).
- [6] Kaneko, S., Kondo, T., and Miyamoto, A., "Robust matching of 3D contours using iterative closest point algorithm improved by M-estimation," *Pattern Recognition* **36**(9), 2041–2047 (2003).
- [7] Amberg, B., Romdhani, S., and Vetter, T., "Optimal step nonrigid ICP algorithms for surface registration," in [*IEEE Conference on Computer Vision and Pattern Recognition*], 1–8, IEEE Computer Society, Minnesota, USA (June 2007).
- [8] Feldmar, J. and Ayache, N., "Locally affine registration of free-form surfaces," in [*IEEE Conference on Computer Vision and Pattern Recognition*], 496–501 (June 1994).
- [9] Kambhamettu, C., Goldgof, D., and He, M., "On a study of invariant features in nonrigid transformations," in [*IEEE Workshop on Qualitative Vision*], 118–127 (June 1993).
- [10] Kambhamettu, C., Goldgof, D., and He, M., "Determination of motion parameters and estimation of point correspondences in small nonrigid deformations," in [*IEEE Conference on Computer Vision and Pattern Recognition*], 943–946 (June 1994).
- [11] Laskov, P. and Kambhamettu, C., "Comparison of 3D Algorithms for Non-rigid Motion and Correspondence Estimation," in [*British Machine Vision Conference*], 273–282 (Sept. 2001).
- [12] Laskov, P., *Extensions of Differential-Geometric Algorithms for Estimation of 3D Non-Rigid Motion and Correspondence*, PhD thesis, University of Delaware (2001).
- [13] Kambhamettu, C., Goldgof, D., He, M., and Laskov, P., "3D nonrigid motion analysis under small deformations," *Image and Vision Computing* **21**(3), 229–245 (2003).
- [14] Laskov, P. and Kambhamettu, C., "Curvature-based algorithms for nonrigid motion and correspondence estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**(10), 1349–1354 (2003).
- [15] Li, M., Kambhamettu, C., and Stone, M., "Spline-based motion recovery for 3D surfaces using nonrigid shape properties," in [*IEEE Conference on Computer Vision and Pattern Recognition Workshops*], 23–31 (June 2004).
- [16] Li, M., Kambhamettu, C., and Stone, M., "Nonrigid motion recovery for 3D surfaces," *Image Vision Comput.* **25**(3), 250–261 (2007).
- [17] Faugeras, O. and Hebert, M., "The representation, recognition, and locating of 3-D objects," *Int. J. Rob. Res.* **5**(3), 27–52 (1986).
- [18] Horn, B., "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A* **4**, 629–642 (Apr. 1987).
- [19] Umeyama, S., "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**, 376–380 (Apr. 1991).

- [20] Kambhamettu, C., *Nonrigid motion analysis under small deformations*, PhD thesis, University of South Florida (1994).
- [21] Goldfeather, J. and Interrante, V., “A novel cubic-order algorithm for approximating principal direction vectors,” *ACM Trans. Graph.* **23**(1), 45–63 (2004).
- [22] Combès, B. and Prima, S., “Prior affinity measures on matches for ICP-like nonlinear registration of free-form surfaces,” in [*IEEE International Symposium on Biomedical Imaging*], 370–373 (June 2009).
- [23] Yeo, B., Sabuncu, M., Vercauteren, T., Ayache, N., Fischl, B., and Golland, P., “Spherical Demons: Fast Surface Registration,” in [*Medical Image Computing and Computer Assisted Intervention*], 745–753 (2008).
- [24] Durrleman, S., Pennec, X., Trounev, A., and Ayache, N., “Sparse Approximation of Currents for Statistics on Curves and Surfaces,” in [*Medical Image Computing and Computer Assisted Intervention*], 390–398 (2008).