Registration and Reconstruction

Rigid and Affine Registration of Smooth Surfaces using Differential Properties

Jacques Feldmar and Nicholas Ayache

INRIA SOPHIA, Projet EPIDAURE
2004 route des Lucioles, B.P. 93
06902 Sophia Antipolis Cedex, France.
Email : Jacques.Feldmar@sophia.inria.fr

Abstract. Recently, several researchers ([BM92], [Zha93], [CM92], [ML92], [CLSB92]) have proposed very interesting methods based on an iterative algorithm to rigidly register surfaces represented by a set of 3d points, when an estimate of the displacement is available. In this paper, we propose to introduce differential informations on points to extend this algorithm. First, we show how to efficiently use curvatures to superpose principal frame at possible corresponding points in order to find the needed rough estimate of the displacement. Then, we explain how to extend this algorithm to look for an affine transformation between two surfaces. We introduce differential informations in points coordinates : this allows us to match locally similar points. We show how this differential information is transformed by an affine transformation. Finally, we introduce curvatures in the best affine transformation criterion and we minimize it using extended Kalman filters. All this extensions are illustrated with experiments on various real biomedical surfaces : teeth, faces, skulls and brains.

1 Introduction

In this paper, we are interested in surface matching. When two surfaces represent the same object, it is often useful to superpose them. For instance, this is important for the medical diagnosis to compare images acquired at different times or coming from different modalities. See the article of Lisa Gottesfeld Brown [Bro92] for a review of the existing techniques. Our work is an extension of an iterative algorithm used in [BM92], [Zha93], [ML92], [CM92] and [CLSB92]. This algorithm is described in section 2.1 and is called "the iterative algorithm" in this article. Of course, classical techniques to register non-smooth objects have influenced us. The book of Grimson ([Gri90]) is a very good review of these. The geometric hashing method proposed in [GA92] to match crest lines and the method based on mechanic of [MR92] have also influenced us.

When the two surfaces do not come from the same object, but from objects of the same class (for example two faces) it is very useful too to find the match. An example of application is to match a brain with an atlas in order to find abnormalities or to segment the brain into anatomical regions. See again [Bro92]

Lecture Notes in Computer Science, Vol. 801
Jan-Olof Eklundh (Ed.)
Computer Vision – ECCV '94
© Springer-Verlag Berlin Heidelberg 1994

for a review. We just quote [CAS92] because their use of curvature to track points on deformable objects has influenced us.

We propose to introduce differential information to extend the iterative algorithm. In section 2, we first present this algorithm (2.1). Then we present our method to efficiently find the requisite initial estimate (2.2). In section 3, we explain how we have extended the iterative algorithm in order to find a good affine transformation. We first introduce differential informations in point coordinates (3.2). Then we modify the definition of the best affine transformation (3.3). Finally, we present results on real data (3.4).

2 Computing the rigid displacement

2.1 The iterative algorithm

We now briefly describe the iterative algorithm. (refer to the original papers for details). The goal is to find the rigid displacement $(\mathbf{R}, \mathbf{t})$[1] to superpose two surfaces, S_1 on S_2, given a rough estimate $(\mathbf{R}_0, \mathbf{t}_0)$ of this rigid displacement. Each surface is described by a set of 3d-points. The algorithm consists of two iterated steps, each iteration i computing a new estimation $(\mathbf{R}_i, \mathbf{t}_i)$ of the rigid displacement.

(1) The first step builds a set $Match_i$ of pairs of points. The construction is very simple : for each point M on S_1, a pair (M, N) is added to $Match_i$, where N is the closest point on S_2 to the point $\mathbf{R}_{i-1}M + \mathbf{t}_{i-1}$. To compute the closest point, different methods are proposed but one can use for example the distance map method [Dan80].

(2) The second step is just the least square evaluation of the best rigid displacement $(\mathbf{R}_i, \mathbf{t}_i)$ to superpose the pairs of $Match_i$ (see for example [FH86] for the quaternion method).

The termination criterion depends on the authors : the algorithm stops either when a) the distance between the two surfaces is below a fixed threshold, b) the variation of the distance between the two surfaces at two successive iterations is below a fixed threshold or c) a maximum number of iterations is reached.

This algorithm is very efficient and finds the right solution when the initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$ of the rigid displacement is "not too bad" and when each point on S_1 has a correspondent on S_2. But, in practice, this is often not the case and we have to find the prior estimate $(\mathbf{R}_0, \mathbf{t}_0)$ and to deal with occlusion.

2.2 Finding the initial rigid displacement

We use differential informations to get it. The surfaces we have to superpose come from techniques described in [TG92], [Gué93]. So, for each point M on the surface, we know the principal curvatures and the principal frame.

[1] A rigid displacement $(\mathbf{R}, \mathbf{t})$ maps each point M to $\mathbf{R}M + \mathbf{t}$ where $\mathbf{R}$ is a 3x3 rotation matrix and $\mathbf{t}$ a translation vector.

In the ideal case, because principal curvatures are invariant under rigid displacement, given a point M on S_1 with principal curvatures (k_1, k_2), a point N on S_2 must have the same curvatures to be a possible correspondent. Moreover, if the pair (M, N) is a good match, then the rigid displacement which superposes S_1 on S_2 is also the one which superposes the principal frames attached to M and N respectively on S_1 and S_2. Hence, in the ideal case, the following algorithm would be very efficient : **(1)** choose a point M on S_1, **(2)** compute the set $SameCurvature(M)$ of points on S_2 which have the same curvatures as M, **(3)** for each point N in $SameCurvature(M)$, compute the rigid displacement corresponding to the superposition of the two principal frames and stop when this rigid displacement exactly superposes S_1 on S_2.

But in practice, the two surfaces cannot be exactly superposed, and there is the principal frame orientation problem. To deal with imprecision on curvatures, we register the points of S_2 in a hash table or in a kd-tree (see [PS85]) indexed by the two principal curvatures. This way, given a point M on S_1, with curvatures (k_1, k_2), we can quickly find the set of points $CloseCurvature(M)$ on S_2 whose curvatures are close to (k_1, k_2). Then, we apply the following algorithm :

1. we randomly choose a point M on the surface S_1
2. we compute the set $CloseCurvature(M)$
3. for each point N in $CloseCurvature(M)$, we compute the rigid displacements corresponding to the superposition of the two principal frames. If $R_1 = (M, \mathbf{e}_{11}, \mathbf{e}_{21}, \mathbf{n}_1)$ is the principal frame at point M and $R_2 = (N, \mathbf{e}_{12}, \mathbf{e}_{22}, \mathbf{n}_2)$ the principal frame at point N, we compute two rigid displacements d and d'. d corresponds to the superposition of R_1 on R_2. d' corresponds to the superposition of R_1 on R_2', where $R_2' = (N, -\mathbf{e}_{12}, -\mathbf{e}_{22}, \mathbf{n}_2)$. We have to compute these two rigid displacements because R_2 and R_2' are both direct, and there is no way to choose between them[2].
4. we now estimate the ratio of the number of points on the transformed surface $\mathbf{R}S_1 + \mathbf{t}$ which have their closest point on S_2 below a given distance, on the number of points of the surface S_1 to check if either d or d' reaches our termination criterion. First, we randomly choose a subset S_1' of points on S_1. Then, for each point P in S_1', we compute the closest point Q to $\mathbf{R}P + \mathbf{t}$ on S_2 and if $\|\mathbf{R}P + \mathbf{t} - Q\|$ is below a given threshold δ, then we add the pair (P, Q) in a set $Pair\_ok$. Finally, if the ratio $|Pair\_ok|/|S_1'|$ is bigger than $Threshold$ $(0 < Threshold < 1)$, we decide that $(\mathbf{R}, \mathbf{t})$ is a good estimate of the rigid displacement which superposes S_1 on S_2 and we stop the algorithm. If neither d nor d' reaches the termination criterion, then we return to point 1.

The parameter δ is the estimation of the maximal distance between one point on the surface S_1 and its closest point on S_2 after the best rigid registration (depends on the noise). $Threshold$ is the estimation of the ratio of the number of points on S_1 which have a correspondent on S_2, on the number of points

[2] Note that we are able to orient the normals because, in practice, we know the interior and the exterior of the objects.

on S_1 (depends on the occlusion). In practice, for our problems, it is not difficult to choose them and a good solution is found after a very small number of iterations (two or three). Typical surfaces we work with have around 10000 points. $CloseCurvature(M)$ is the set of points N on S_2 whose the principal curvatures (k'_1, k'_2) are such that $((k'_1 - k_1)^2 + (k'_2 - k_2)^2)^{1/2} < Dim/20$ where $Dim = \max(Dim\_k_1, Dim\_k_2)$ (Dim_k_1 (respectively Dim_k_2) is the difference between the maximum and the minimum value of k_1 (respectively k_2) in S_2). For S'_1, we randomly choose 5% of points of S_1. For example, the figure 1 (left) shows the initial estimate found for teeth data. We have chosen $Threshold = 0.8$ and $\delta = D/30$ where D is the largest surface diameter. The rigid displacement is found after two iterations in less than twenty seconds on a DEC 5000 workstation.

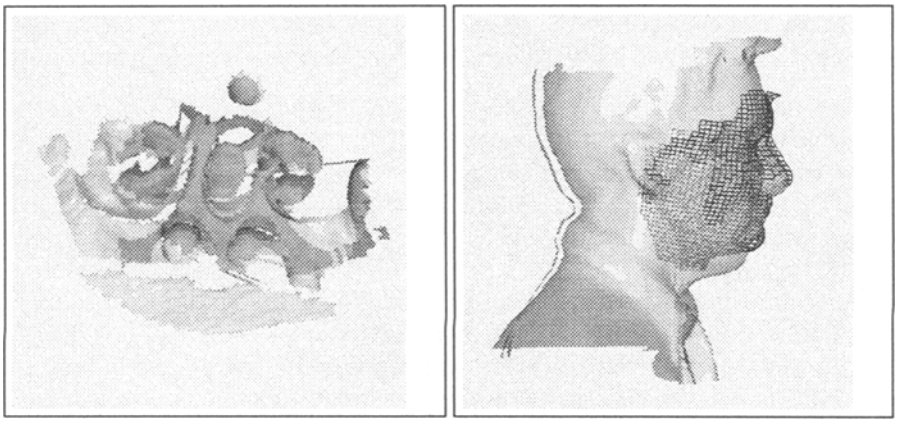

Fig. 1. Left : the rough estimate of the rigid displacement for teeth data acquired by Sopha-Bioconcept using a laser technique. One surface is dark, the other one is bright. **Right** : The best rigid displacement found between two faces of two different people. They have been acquired using a Cyberware machine. One surface is transparent, the other one is represented by lines. When the lines are dark, they are in front of the transparent surface, when they are not, they are behind.

This algorithm to find initial estimates associated with the iterative algorithm yields to an efficient framework to find very accurate rigid displacements. When the two surfaces are not complete because of occlusion, we just make use of covariance matrices and generalized Mahalanobis distances to decide if a point on S_1 has a correspondent on S_2 or not, as described in [Aya91]. For example, for teeth data of figure 1, the final displacement is such that 75% of points on the transformed surface have their closest point on the other surface at a distance lower than 0.75% of the largest surface diameter. This approximatively corresponds to the occlusion.

3 Non rigid matching of two different surfaces

When the two surfaces do not come from the same object, but from objects of the same kind, the framework described in section 2 is robust enough to find the best rigid displacement to register them. For example, figure 1 (right) shows it for two faces of two different people. In order to improve the superposition, a natural extension from rigid transformations to non rigid ones is the search for an unconstrained affine transformation $(\mathbf{A}, \mathbf{b})$[3].

3.1 Finding an affine transformation

We could use the rigid displacement found as described in the previous section as an initial estimate $(\mathbf{A}_0, \mathbf{b}_0)$ and just modify the second step of the iterative algorithm. It would just be the least square evaluation of the best affine transformation $(\mathbf{A}_i, \mathbf{b}_i)$ to superpose the pairs of $Match_i$. This evaluation is a classical least square problem.

But in practice, this very simple algorithm does not find a good solution : the similarities on the two surfaces do not tend to be brought nearer. Moreover, another major problem occurs with it. It often does not tend to a stable solution : when the transformed surface $\mathbf{A}S_1 + \mathbf{b}$ becomes very small or very flat, the criterion is minimized and nothing in the algorithm tends to avoid it. Especially, when $\mathbf{A}$ is the null matrix and $\mathbf{b}$ corresponds to a point on S_2, the criterion vanishes. This is for example what happens with the faces of figure 1. To avoid these two problems, we describe in sections 3.2 and 3.3 the modifications we bring to the original iterative algorithm (section 2.1), respectively to step 1 and step 2.

3.2 Matching locally similar points

Because points belong to surfaces, we would like that points with local similarity of shape tend to be matched in step 1 of the iterative algorithm. Because a point on a surface is very locally described by the principal curvatures and the principal frame, adding coordinates corresponding to this differential informations to the three spatial coordinates, we obtain the desired effect. In our formulation, surface points are no longer 3d points : they are 8d points. Coordinates of a point M on the surface S are $(x, y, z, n_x, n_y, n_z, k_1, k_2)$ where (n_x, n_y, n_z) is the normal on S at M, and k_1, k_2 are the principal curvatures. Between two points $M(x, y, z, n_x, n_y, n_z, k_1, k_2)$ and $N(x', y', z', n'_x, n'_y, n'_z, k'_1, k'_2)$ we now define the distance :

$$d(M,N) = (\alpha_1(x-x')^2 + \alpha_2(y-y')^2 + \alpha_3(z-z')^2 + \alpha_4(n_x-n'_x)^2 + \alpha_5(n_y-n'_y)^2 + \alpha_6(n_z-n'_z)^2 + \alpha_7(k_1-k'_1)^2 + \alpha_8(k_2-k'_2)^2)^{1/2}$$

where α_i is the inverse of the difference between the maximal and minimal value of the i^{th} coordinate of points in S_2. Using this new definition of the distance, the

[3] An affine transformation $(\mathbf{A}, \mathbf{b})$ maps each point M to $\mathbf{A}M + \mathbf{b}$ where $\mathbf{A}$ is a 3x3 matrix and $\mathbf{b}$ a translation vector.

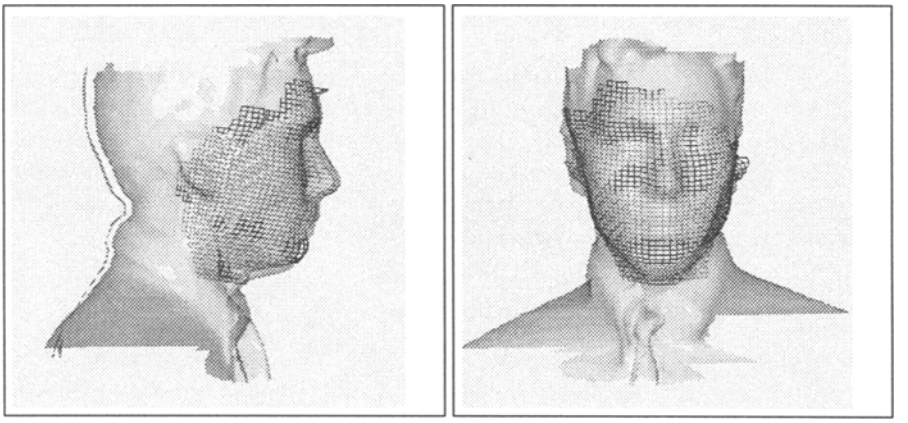

Fig. 2. The final affine transformation between the two faces. This has to be compared with the rigid displacement of figure 1 (right). One can see that the nose, the mouth and the chin are now much closer.

closest point to P on S_2 is a compromise between the 3d distance, the difference of normal orientation[4] and the difference of curvatures.

But this new definition of points coordinates introduces an interesting problem. At step 1 of the iterative algorithm, we have to compute $ClosestPoint(\mathbf{A}_i M + \mathbf{b}_i)$ where M is a point on S_1. Hence, we have to compute the new coordinates $(x', y', z', n'_x, n'_y, n'_z, k'_1, k'_2)$ of $\mathbf{A}_i M + \mathbf{b}_i$, where $(x', y', z')^t = \mathbf{A}_i(x, y, z)^t + \mathbf{b}_i$, (n'_x, n'_y, n'_z) is the normal on the transformed surface $\mathbf{A}_i S_1 + \mathbf{b}_i$ at point (x', y', z'), and k'_1 and k'_2 are the principal curvatures. In fact, because we need this result in section 3.3 we show in [FA94b] that :

proposition 1 :

When a surface S is transformed into a surface $\mathbf{A}S + \mathbf{b}$ by an affine transformation $(\mathbf{A}, \mathbf{b})$, the principal frame and the curvatures at point $\mathbf{A}M + \mathbf{b}$ on $\mathbf{A}S + \mathbf{b}$ depend only on the principal frame and the curvatures at point M on S.

More precisely, there exists a parameterization of $\mathbf{A}S_1 + \mathbf{b}$ such that, denoting E', F', G' the coefficients of the first fundamental form at point $\mathbf{A}M + \mathbf{b}$ on $\mathbf{A}S_1 + \mathbf{b}$, e', f', g' the coefficients of the second fundamental form and $(M, \mathbf{e}_1, \mathbf{e}_2, \mathbf{n})$ the principal frame at point M on S_1, we have :

$$\begin{cases} E' = \mathbf{A}\mathbf{e}_1.\mathbf{A}\mathbf{e}_1, & F' = \mathbf{A}\mathbf{e_1}.\mathbf{A}\mathbf{e}_2, & G' = \mathbf{A}\mathbf{e}_2.\mathbf{A}\mathbf{e}_2 \\ e' = \dfrac{det(\mathbf{A})k_1}{\|\mathbf{A}\mathbf{e}_1 \wedge \mathbf{A}\mathbf{e}_2\|}, & f' = 0, & g' = \dfrac{det(\mathbf{A})k_2}{\|\mathbf{A}\mathbf{e}_1 \wedge \mathbf{A}\mathbf{e}_2\|} \end{cases} \quad (1)$$

[4] Of course, only two parameters are necessary to describe the orientation of the normal (for example the two Euler angles). But we use (n_x, n_y, n_z) because, this way, the Euclidean distance really reflects the difference of orientation between the normals (that is not the case with the Euler angles) and we can use the kd-trees to find the closest point as explained later.

Because we know the coefficients of the fundamental forms of the transformed surface $\mathbf{A}S_1 + \mathbf{b}$ at point $\mathbf{A}M + \mathbf{b}$, we can compute the coordinates $(x', y', z', n'_x, n'_y, n'_z, k'_1, k'_2)$ (see [dC76]).

Just a problem remains to introduce this new definition of points coordinates : computing $ClosestPoint(\mathbf{A}_i M + \mathbf{b}_i)$. In 8d, we cannot use the technique described in [Dan80] as in the 3d case. The distance map would be much too big. We use the kd-tree technique as proposed in [Zha93] for the 3d case. This takes more time than before. Each iteration takes 45 seconds (CPU time) instead of 9 seconds when surfaces have 7000 points. To improve the performances, it is possible to work on a subset of points of surfaces. For example, it is possible to extract crest lines points [TG92]. Or simply, we can select a given percentage of points, choosing points which have the highest mean curvature. What is important is that most of the selected points on S_1 have a correspondent on S_2 and that the selected points describe relatively well the surfaces.

3.3 Constraints on the affine transformation

We have now to focus on a major drawback in the previous search for the affine transformation which best superposes S_1 on S_2. The criterion we minimize at step 2 of the iterative algorithm can always vanish as explained in section 3.1. To avoid this problem, we propose to define a new criterion. Let $(x, y, z, k_1, k_2, \mathbf{e}_1, \mathbf{e}_2)_k$ be the 3d coordinates, principal curvatures and principal directions of points M_k on S_1. Let $(x', y', z', k'_1, k'_2)_k$ be the 3d coordinates and principal curvatures of points $\mathbf{A}M_k + \mathbf{b}$ on the transformed surface. We call $\mathbf{g}$ the function which associates $(x', y', z', k'_1, k'_2)_k$ to $((x, y, z, k_1, k_2, \mathbf{e}_1, \mathbf{e}_2)_k, \mathbf{A}, \mathbf{b})$. The existence of this function is a consequence of the proposition 1 and we use the equations (1) to compute it. The new criterion we propose to minimize at step 2 of the algorithm is :

$$\sum_{(M_k, N_k) \in Match_i} p_k \|\mathbf{g}((x, y, z, k_1, k_2, \mathbf{e}_1, \mathbf{e}_2)_k, \mathbf{A}, \mathbf{b})) - N_k\|^2 \qquad (2)$$

where the coordinates of N_k are $(x'', y'', z'', k''_1, k''_2)_k$: the 3d coordinates and the two principal curvatures. This new criterion measures both the 3d distance and the difference of curvature between S_2 and the transformed surface $\mathbf{A}S_1 + \mathbf{b}$. Moreover, the coefficients p_k allow us to increase the importance in the criterion of the match for high curvature points because they seem to have a strong anatomical meaning ([Aya93]). These coefficients can be, for example, the mean curvature or $max(|k_1|, |k_2|)$.

We use the extended Kalman filter formalism (EKF) to minimize this new criterion (2). The details are given in [FA94b]. The only difficulty is to derive $\mathbf{g}$ with respect to $\mathbf{A}, \mathbf{b}$ and (x, y, z, k_1, k_2). In practice, we made numerous experiments and even if the minimized criterion is nonlinear, the minimization works very well. Using the new definition of the closest point (section 3.2) at step 1 and this new criterion at step 2, the modified iterative algorithm find good and stable solutions. Figure 2 shows the affine transformation found for the faces of figure

1. The solution is found after ten iterations in about 7.5 minutes (CPU time) on a DEC 5000 workstation[5]. It is to compare with the rigid transformation of figure 1. The chin, the mouth, the nose and the eyebrows of the two faces are now much closer. Note that the surfaces are now so close, that each one alternatively tends to appear in front of the other. To quantitatively evaluate the error, we have computed the average distance between a point of the transformed surface $\mathbf{A}S_1 + \mathbf{b}$ and its closest point on S_2. Setting the largest surface diameter to u, the rigid transformation yields to an average distance of $0.0193u$ when we use the 3d Euclidean norm to compute both the distance and the closest point, whereas the affine transformation (figure 2) yields to an average distance of $0.0152u$. This is a 22% improvement.

3.4 Results

In this section, we present results on skull data (figure 3) and brain data (figure 4). Of course, the two surfaces come from two different patients. The left

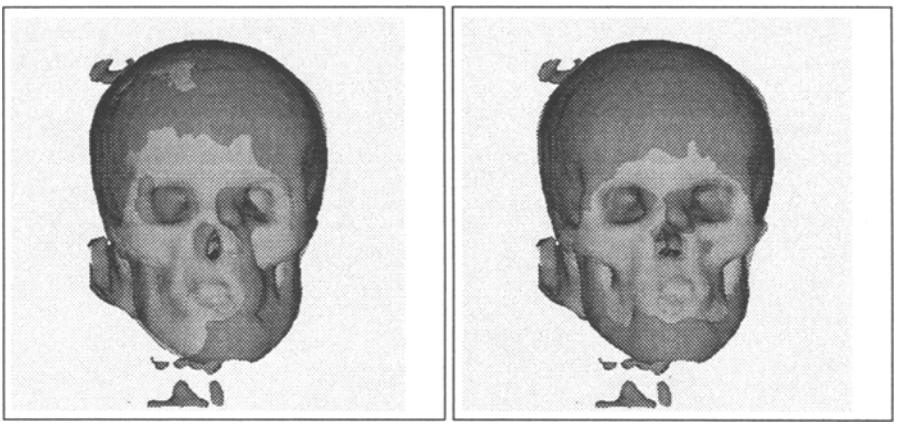

Fig. 3. The superposition of two skulls. The surfaces have been extracted in X-ray scanner images (New-York University Medical Center, Court Cutting). One surface is dark, the other one is bright. **Left** : result using a rigid displacement. **Right** : result using an affine transformation. One can observe that the hole corresponding to the nose and the orbits are not well registered with the rigid displacement, whereas they are with the affine transformation.

images show the rigid displacement found between the two surfaces whereas the right images show the affine transformation. For the skull example, setting the largest surface diameter to u, the error is $0.0120u$ for the rigid displacement and

[5] This time could be drastically reduced selecting on the surfaces characteristic points.

$0.0098u$ for the affine transformation. This is a 22% improvement. The affine transformation is found after 8 iterations in about 8 minutes on a DEC 5000 workstation.

For the brain example, the error is $0.011u$ for the rigid displacement and $0.0091u$ for the affine transformation. This is a 18% improvement. The affine transformation is found after 12 iterations in about 7 minutes.

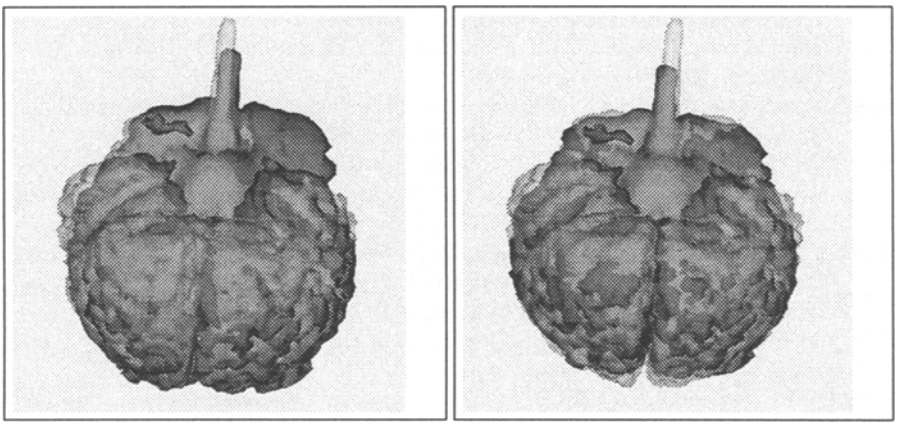

Fig. 4. The superposition of the two brains. The surfaces have been extracted in MRI images (Brigham and Women's Hospital of Boston). **Left** : result using a rigid displacement. **Right** : result using an affine transformation. The brightest surface is transparent. Hence, when the other surface is in front, it appears dark, and when it is behind, it appears less dark. One can observe that the interhemispherical fissure is not well registered with the rigid displacement, whereas it is with the affine transformation.

4 Conclusion

We have described and implemented a complete framework which is fast and robust to find very accurate rigid displacements between smooth surfaces. We have extended this framework to deal with affine transformations. We illustrated that the best affine transformation is generally found and that it brings the surfaces much nearer than a rigid displacement. We believe that this is a good starting point for more local non rigid registration schemes like the ones reviewed in [Bro92]. Moreover, an extension to locally affine transformations is presented in [FA94a].

Acknowledgements

We are very grateful to A. Gourdon, H. Delingette and J.P. Thirion for their help. Thanks are also due to people who gave data to us. This work was supported in

part by a grant from **Digital Equipment Corporation** and by the European Basic Research Action **VIVA** (Esprit project).

References

[Aya91] N. Ayache. *Artificial Vision for Mobile Robots — Stereo-Vision and Multisensory Perception.* Mit-Press, 1991.

[Aya93] N. Ayache. Volume image processing. results and research challenges. Technical Report 2050, INRIA, 1993.

[BM92] Paul Besl and Neil McKay. A method for registration of 3−D shapes. *PAMI*, 14(2):239–256, February 1992.

[Bro92] Lisa Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–375, December 1992.

[CAS92] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *ECCV 1992*, Santa Margherita Ligure, Italy, 1992.

[CLSB92] G. Champleboux, S. Lavallée, R. Szeliski, and L. Brunie. From accurate range imaging sensor calibration to accurate model-based 3−D object localization. In *CVPR*, Urbana Champaign, 1992.

[CM92] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[Dan80] P.E. Danielsson. Euclidean distance mapping. *Computer Graphics and Image Processing*, 14:227–248, 1980.

[dC76] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, 1976.

[FA94a] J. Feldmar and N. Ayache. Locally affine registration of free-form surfaces. In *CVPR 94*, Seattle, 1994.

[FA94b] J. Feldmar and N. Ayache. Rigid, affine and locally affine registration of free-form surfaces. Technical report, INRIA, 1994.

[FH86] O. Faugeras and M. Hébert. The representation, recognition and locating of 3*d* objects. *Int. J. Robotics Res*, 5(3):27–52, 1986.

[GA92] A. Guéziec and N. Ayache. Smoothing and matching of 3−D-space curves. In *ECCV 1992*, Santa Margherita Ligure, Italy, 1992.

[Gri90] W. Grimson. *Object Recognition by Computer: The role of geometric constraints.* MIT Press, 1990.

[Gué93] André Guéziec. Large deformable splines, crest lines and matching. In *ICCV 93*, Berlin, 1993.

[ML92] Yau H.-T. Menq, C.-H. and G.-Y. Lai. Automated precision measurement of surface profile in cad-directed inspection. *IEEE Trans. RA*, 8(2):268–278, 1992.

[MR92] G. Malandain and J.M. Rocchisani. Registration of 3−D medical images using a mechanical based method. In *EMBS 92, Satellite Symposium on 3−D Advanced Image Processing in Medicine*, Rennes, France, 1992.

[PS85] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry, an Introduction.* Springer Verlag, 1985.

[TG92] J.P. Thirion and A. Gourdon. The 3−D marching lines algorithm and its application to crest lines extraction. Technical Report 1672, INRIA, 1992.

[Zha93] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surface. *Int. Journal of Computer Vision*, 1993.