INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

New Feature Points based on Geometric Invariants for 3D Image Registration

Jean-Philippe THIRION

N° 1901

Mai 1993

PROGRAMME 4

Robotique,
image
et vision

ISSN 0249-6399

New Feature Points based on Geometric Invariants for 3D Image Registration

Jean-Philippe THIRION *

Programme 4 — Robotique, image et vision
Projet Epidaure **

Rapport de recherche n ° 1901 — Mai 1993 — 31 pages

Abstract: We introduce in this paper a new type of feature points of 3D surfaces, based on geometric invariants. We call this new type of feature points the *extremal points* of the 3D surfaces, and we show that the relative positions of those 3D points are invariant according to 3D rigid transforms (rotation and translation). We show also how to extract those points from 3D images, such as Magnetic Resonance images (MRI) or Cat-Scan images, and also how to use them to perform precise 3D registration. Previously, we described a method, called the Marching Lines algorithm, which allow us to extract the extremal lines, which are geometric invariant 3D curves, as the intersection of two implicit surfaces : the extremal points are the intersection of the extremal lines with a third implicit surface. We show an application of the extremal points extraction to the fully automatic registration of two 3D images of the same patient, taken in two different positions, which exhibits the remarkable accuracy and robustness of the extracted extremal points.

Key-words: 3D Image Processing, Differential Geometry, Geometric Invariant, Iso-Surface, crest line, Marching Line.

(Résumé : tsvp)

*Email: jean-philippe.thirion@sophia.inria.fr
**http://zenon.inria.fr:8003/Equipes/EPIDAURE-eng.html

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

De nouveaux points caractéristiques basés sur l'utilisation d'invariants géométriques pour le recalage d'images 3D

Résumé :

Nous décrivons ici un nouveau type de points caractéristiques, définis à l'aide d'invariants géométriques, et que l'on peut extraire d'images tridimensionnelles telles que celles fournies par la Résonance Magnétique (IRM), ou le scanner à rayons X. Nous appelons ces nouveaux points les *Points Extrémaux* des images 3D, et nous montrons que la position relative de ces points 3D est invariante par transformation géométrique rigide (rotations et translations). Nous montrons comment extraire ces points, et comment les utiliser pour le recalage précis d'images 3D. Précédemment, nous avions décrit un algorithme, le "Marching Lines", nous permettant d'extraire les lignes de crêtes, qui sont des lignes invariantes par transformation rigide, en tant qu'intersection de deux surfaces implicites. Les Points Extrémaux sont les intersections de ces lignes de crêtes avec une troisième surface implicite que nous décrivons ici. Enfin, nous montrons une application de l'extraction des Points Extrémaux au recalage de deux images 3D du même patient, prises dans deux positions différentes, et qui montre la très grande précision et robustesse de notre méthode.

Mots-clé : image 3D, géométrie différentielle, invariant géométrique, isosurface, ligne de crête, Marching Lines.

1 Introduction

Our work takes its origin in the search of stable features to perform the automatic registration of three-dimensional images. We were looking for a method which would have been as much as possible independent from the subject scanned. For example, for the case of 3D medical images, methods based on anatomical invariants were prohibited, because we would have designed then a registration software which would have worked for, say, the head examination, but not for the liver.

We found the solution of our problem in the use of the geometric invariants of 3D surfaces. We get very good results for the automatic registration of high quality 3D images of the same subject, taken in two different positions and with the same acquisition device. However there is still more work to do to perform reliably the registration with 3D images coming from two different devices (inter-modality registration), such as for example between a Magnetic Resonance Image (MRI), and a Single Photon Emission Computed Tomography (SPECT) Image. This is true also for the registration of two different patients (inter-patient registration).

We present here a fully automatic method of registration based on a new set of invariant feature points, that we call the *extremal points* of 3D images, and we present experiments for the mono-modality, mono-patient registration case.

For the majority of previously proposed methods, the first step is to extract the surface of the scanned object, and then to use this surface for the matching. The goal is to reduce the total amount of information to process to a 2D variety of point : the surface. These kind of approaches can be found in [14], [2], [1], [10], [7].

On the contrary of those approaches, based on surface matching, we have concentrated our work on the extraction of characteristic lines and feature points. In previous works (see [15], [16]), we have shown how to extract automatically the crest lines from 3D images, and how to use them to perform the automatic registration (see [6]). The crest lines are a more compact representation of the information of a 3D image than the surface, because they form a 1D variety of points. This lessens many of the ambiguities of the surface matching, ensuring a more stable solution, and also, by reducing the amount of information to process, it allows the study of more evolved registration algorithms (for example, Geometric Hashing, see [6]). Since then, we have

used this method to perform the automatic registration of a wide panel of 3D images (MRI or Cat-Scan).

The present paper is another step in our quest of stable features, and we go now from the crest lines (a 1D variety of points), to the extremal points (a 0D variety). The extremal points are a scattered set of stable points throughout the 3D image.

In the next sections, we first define the extremal points, and explain why they are geometric invariant points. We show then how to extract those points from 3D images with a modified version of the Marching Lines algorithm. Once that the extremal points are extracted, from two different images of the same subject, we present briefly a registration method which is simple, fast and reliable, to match the two sets of points. This registration method makes also use of the geometric invariants associated to the extremal points. We show then experimental results with the registration of two real images, which exhibits the high precision of both the extracted feature points and the registration method.

2 Definitions

Let us first recall briefly some results of differential geometry about surface curvatures (a good introduction to those notions can be found in Do Carmo [3] or in Koenderink [9]).

2.1 Basic notions

At any point P of a 3D surface, an infinite set of curvatures can be defined. More precisely, associated with any direction $\vec{t}$ in the tangent plane of the surface, one can define a directional curvature $k_{\vec{t}}$. This curvature is in fact the curvature of the 3D curve defined by the intersection of the plane $(P, \vec{t}, \vec{n})$ where $\vec{n}$ is normal to the surface, with the surface.

Except for the points where this curvature $k_{\vec{t}}$ is the same for all the directions $\vec{t}$, which are called *umbilic points*, the total set of curvatures can be described with only two privileged directions, $\vec{t}_1$ and $\vec{t}_2$, and two associated curvature values, $k_1 = k_{\vec{t}_1}$ and $k_2 = k_{\vec{t}_2}$, which are called respectively the principal directions and associated principal curvatures of the surface at point P, as shown in figure 1. Those two principal curvatures are the extrema of all

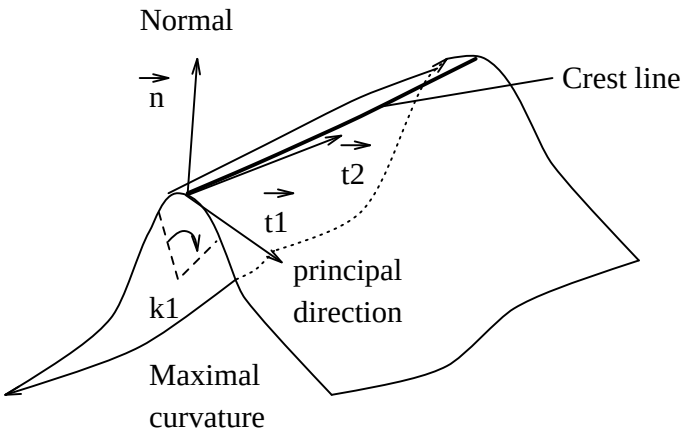

Figure 1: Differential geometry of surfaces

the surface curvatures defined at point P, and (except for umbilic points) one of those two is maximal in absolute value : we call it the *largest curvature*, in order not to be mistaken with the maximal curvature (let say that it is k_1). We simply call *second (principal) curvature* the other principal curvature (k_2).

2.2 Extremal lines

The crest lines are intuitively the loci of the surface where the "curvature" is locally maximal. More precisely, we define them as the loci of the surface where the largest curvature, k_1, is locally maximal (in absolute value), in the associated principal direction $\vec{t_1}$. O. Monga et al. (see [12]) have shown that these points can be defined as the zero-crossing of an extremality function e, which is the directional derivative of k_1 in the direction $\vec{t_1}$, and have proposed a way to characterize them, directly from the voxel values of the 3D image.

We have proposed another method to compute them in [16], for the case of iso-intensity surfaces. Our method is based on the use of the implicit functions theorem. Basically, we have shown that the crest lines can be extracted as the intersection of two implicit surfaces $f = I$ and $e = 0$, where f represents the intensity value of the image, I an iso-intensity threshold, and $e = \vec{\nabla} k_1 \cdot \vec{t_1}$ is

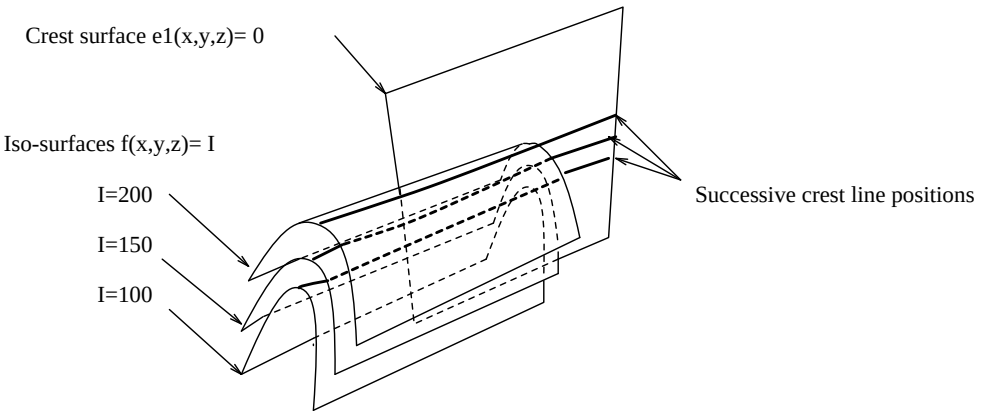

Figure 2: Crest lines as intersection of two implicit surfaces

the extremality function (see figure 2). We have proposed also an algorithm, called the Marching Lines, to automatically extract those crest lines. This algorithm can be used also to overcome some orientation problems (mainly due to the fact that the principal directions are directions, and not oriented vectors), by locally orienting the principal directions, along the extracted lines.

In fact, for each point of the surface, two different extremality coefficients can be defined, corresponding to the two principal curvatures :

$$\begin{aligned} e_1 &= \vec{\nabla} k_1 \cdot \vec{t}_1 \\ e_2 &= \vec{\nabla} k_2 \cdot \vec{t}_2 \end{aligned} \tag{1}$$

Furthermore, we are able, with orientation considerations, to distinguish between the zero-crossings of e_1 and e_2 corresponding to the maxima of the absolute value of the curvature from those corresponding to the minima, by considering the signs of the derivatives of the extremality $\vec{\nabla} e_i \cdot \vec{t}_i$, or more simply, the signs of the extremality for each sides of the zero-crossing. We found experimentally that the maxima (in absolute values) are more stable landmarks than the minima : crests or rifts (maxima) are stable, whereas the loci in a valley where the ground floor is the flattest (minima) are very sensitive to a small perturbation of the data (see figure 9).

We call *extremal lines* all the lines defined as the zero-crossings of either e_1 or e_2. There is therefore 4 major different types of extremal lines, depending of whether the corresponding curvature is the largest or the second one, and whether it is a local maximum or minimum. Furthermore, the signs of the largest and second curvatures help to distinguish between 4 more sub-types of extremal lines, leading to a classification into 16 types. The crest lines are two of them : positive largest curvature maxima ($k_1 > 0$ and $\vec{\nabla} e_1 \cdot \vec{t_1} < 0$) and negative largest curvature minima ($k_1 < 0$ and $\vec{\nabla} e_1 \cdot \vec{t_1} > 0$).

2.3 Extremal points

We define now the extremal points as the intersection of three implicit surfaces : $f = I$, $e_1 = 0$ and $e_2 = 0$, as shown in figure 3.

As we can see with the figure 3, the loci of the extremal points when the iso-value I is changed continuously, are 3D curves (the intersection of the two extremal surfaces $e_1 = 0$ and $e_2 = 0$), which are *intrinsic* to the 3D image, just as the extremal surfaces are. We call *extremal points lines* those 3D curves, whose intersection with a given iso-surface gives the extremal points. The notions of extremal lines and extremal points are closely related to the notion of corner points, in 2D images, as defined in [8], [13], [5]. A remarkable study of the evolution in 2D of those corner points with respect to the scale can be found in [4], chapter 5. We have started similar studies about the influence of the scale on the extremal lines and points in the case of 3D images.

e_1 and e_2 are geometric invariants of the implicit surface $f = I$, that is, are preserved with rigid transforms (rotations and translations of the object). Therefore, the relative positions of the extremal points are also invariants to a rigid transform, i.e. for two different acquisitions of the same subject.

The intuitive interpretation of extremal points is however not straightforward. The extremal lines are 3D curves, for which we are able to compute the curvature (curvature of the 3D lines), but the extremal points are generally not the points of the extremal lines whose curvature is locally maximal. Even if they are not extremal curvature points, the extremal points are very well defined, and there is no reason for their locations along the extremal lines to be less precise that the lines positions themselves, because the precision of the computation of k_1 and k_2 is the same (although the fact that k_1 is the largest curvature gives some privileges to this one).

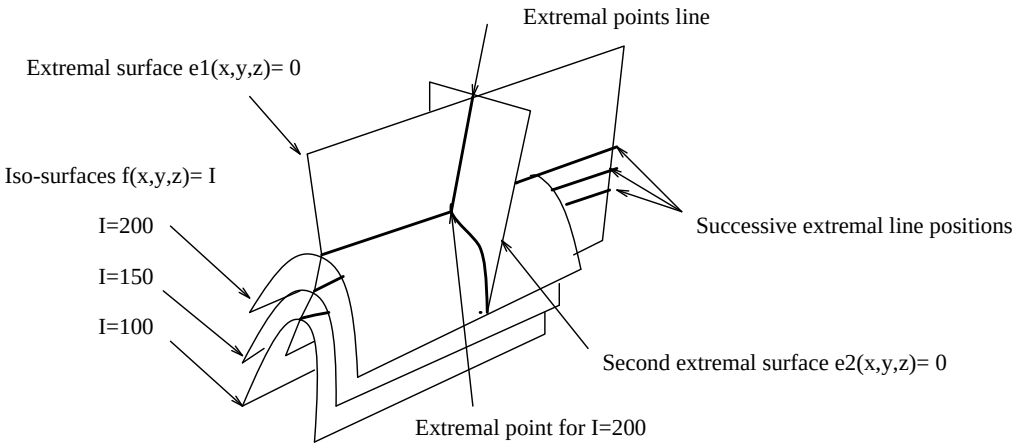

Figure 3: Definition of the extremal points

2.4 Different types of extremal points

The extremal points are the intersections of couples of extremal lines. As there are 16 types of extremal lines, there are also 16 different types of extremal points, depending on the type of extremality (local minimum or maximum) of the extremalities e_1 and e_2, and depending on the signs of k_1 and k_2 (there is unfortunately not 16×16 extremal point types, because one of the two extremal lines corresponds to the largest curvature, and the other one to the second, and the two lines have the same k_1 and k_2 at the extremal point, hence the reduction to only 16 different types).

We use this classification to reduce seriously the complexity of the matching algorithm.

3 The automatic extraction of the extremal points

In practical cases, e_1 and e_2 can be computed for each points of the 3D images with the equations described in the appendix (see also [16]), directly from the differentials of the intensity function of the image f. For range data images (2.5 D images) the computation of e_1 and e_2 is much simpler because of the natural parametrization of the surface and also because the formulae of the first and second fundamental forms of differential geometry, who give the principal directions and curvatures, are generally given for a parametric representation of the surface.

The major point in [16] was to show how to compute those fundamental forms from the implicit representation of the surface. We have shown how to compute those values from the differentials of the image function f, up to order 2 for the fundamental forms, and 3 for the extremality coefficients e_1 and e_2 ($\partial f/\partial x, \partial^2 f/\partial x \partial y, \partial^2 f/\partial x^2 \partial z$, etc ..., a total of 20 different differentials, the formulae are recalled in the appendix).

We compute those differentials with linear filtering, using the convolution of the discrete image with the differentials of the Gaussian function $e^{-r^2/2\sigma^2}$, with $r^2 = x^2 + y^2 + z^2$. The normalization of those filters is however not straightforward, we use the responses to simple polynomials, as proposed by O. Monga et al. We choose the Gaussian function because it is isotropic, a pre-requisite if we are looking for geometric invariants. Different values of σ can be chosen, depending on the level of noise in the 3D images. Changing σ is somewhat equivalent to changing the scale at which we look for extremal lines and points.

The hypothesis that the iso-surfaces are a good representation of the surface of organs for the case of medical image is reasonable : sometimes, the iso-surface can be extracted directly from the 3D image, such as the skin surface in Magnetic Resonance Image (MRI), or the bones in X-ray scanner images. For other soft tissues, such as for the brain surface, a pre-segmentation step is required to isolate the brain from the rest of the data. This can be done with a combination of mathematical morphological operators, filtering, and the search of connected parts. In all cases, the final step of the segmentation is performed with iso-surface techniques.

3.1 Extensive computation of the extremal points

One solution to get the set of extremal points of the 3D image is to compute e_1 and e_2 for all the voxels of the 3D image, and then to consider individually each cubic cell, formed with 8 voxels (8-cell), as shown in figure 4. There are therefore 3 values defined for each vertices of the cube : f, e_1 and e_2. The extremal points in that 8-cell are defined as the intersection of the three implicit surfaces $f = I$, $e_1 = 0$ and $e_2 = 0$. The method varies according to the type of interpolation or convolution function used to extend continuously the three values at the vertices of the cubic cell to the entire cell. The tri-linear interpolation is a good first order approximation.

We have shown in [16] how to extract an approximation based on segments of the crest lines coming through each cell, with some warranties about the topology and the orientation of those reconstructed 3D curves (only f and e_1 were requested). We extend here this method to the computation of the extremal points.

The different steps are summarized in figure 4, for a simple case where the iso-surface of f in the 8-cell can be approximated with only one triangle. The vertices of the triangle are the positions along the edges of the cubic 8-cell (composed with 8 voxel values), for which the linear interpolation of f is I (black dots represent the points where $f \geq I$ and white dots, the points where $f < I$). In that case, we get 3 points : $\{Q_1, Q_2, Q_3\}$, and our algorithm produces one oriented triangle representing the intersection of the iso-surface with the 8-cell.

In the second cube, we show how the extremal line is extracted : we compute the values of e_1 for $\{Q_1, Q_2, Q_3\}$, by linear interpolation of e_1 along the edges of the cubic 8-cell. Black dots represent points where $e_1 \geq 0$ and white dots points where $e_1 < 0$. We compute then the points along the edges of the triangle for which the linearly interpolated value of e_1 along the edge is 0, which gives in that case two points $\{P_1, P_2\}$ which forms an oriented segment. This segment $[P_1, P_2]$ is an approximation of the intersection of the extremal line with the 8-cell.

In [17], we have shown how to perform this operation for any configurations of the values of f and e_1, and in a way that ensures that all the oriented segments produced by this algorithm form 3D oriented curves which are continuous and closed (except for the voxels where f or e_1 are not defined).

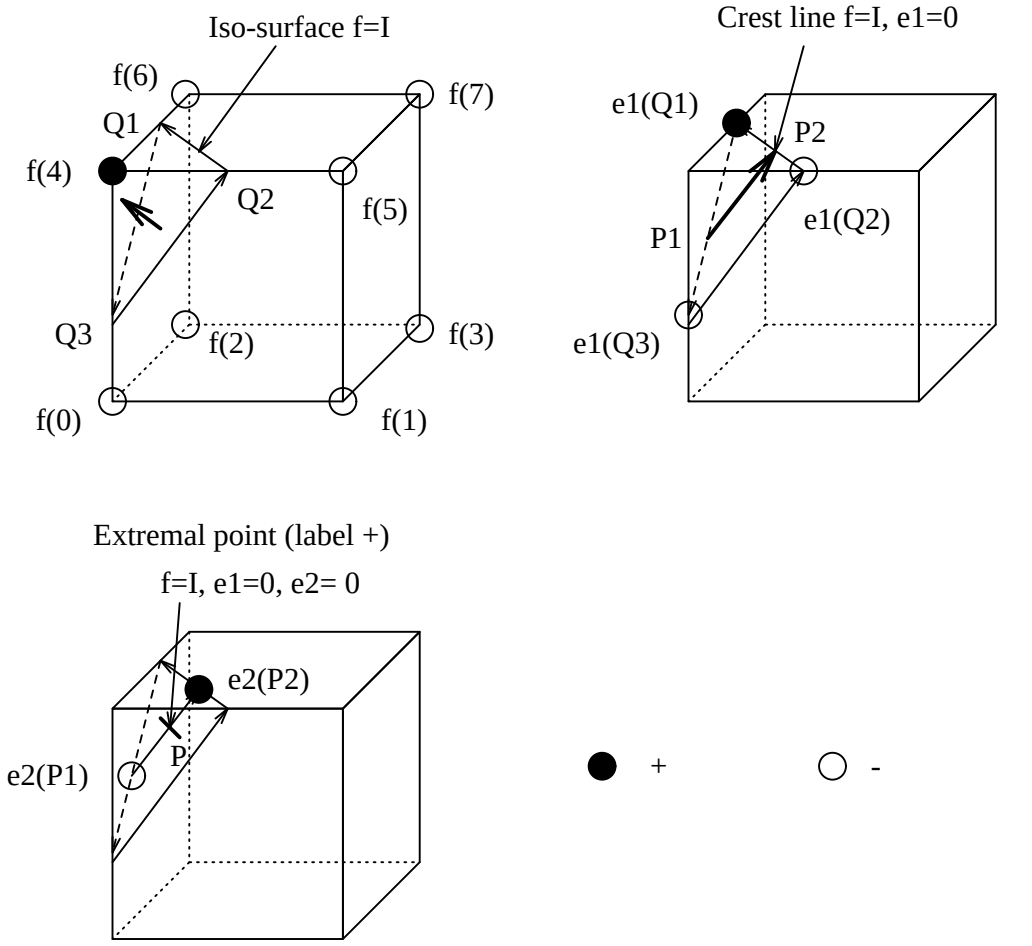

Figure 4: Extraction of the extremal points

The last cube shows how to extract the extremal points from the oriented segments produced by the Marching Lines algorithm. As P_1 and P_2 lie on the faces of the 8-cell, we can compute the value of e_2 for those two points with a bi-linear interpolation of e_2 in those faces. The labels associated to those points are $+$ or "black" if $e_2 \geq 0$ and $-$ or "white" if $e_2 < 0$. If the labels are the same, then we say that there is no extremal point in that segment $[P_1, P_2]$. Otherwise, as it is shown in our particular case, we compute (an approximation of) the extremal point P, being the point along the oriented segment $[P_1, P_2]$ for which the linear interpolation of e_2 (based on $e_2(P_1)$ and $e_2(P_2)$) is 0.

We give a label L to that extremal point P, which is the label of P_2, the end point of the oriented segment. In our case $L = +$: this label is used to distinguish between minimum and maximum points.

We can do the same for each of the cells where the values e_1 and e_2 can be computed (i.e when the gradient is not 0 and the point is not an umbilic point), which generates the total set of extremal points. This constitutes the *extensive* version of the computation of the extremal points. However, the computation of those values is expensive, and we describe now a "randomized" implementation of this method, based on the randomized implementation of the Marching Lines.

3.2 Randomized implementation of extremal points extraction

We start with "seeds", that is, cubic cells of the 3D grid, randomly chosen in the 3D image. A very simple test discards the cells which are not crossed by an iso-surface (the sign of $f - I$ is the same for all the vertices).

Then we compute the values of e_1 for the 8 vertices of the cell. Once again, a simple test discards the cells which are not crossed by a k_1 extremal line (the signs of e_1 is the same for the 8 vertices). If there is an extremal line, we extract it from end to end, using the Marching Lines algorithm (we follow the extremal line "marching" from one cubic cell to the next).

For each cell which is visited, we compute the 8 values of the other extremality e_2 (in fact only the 4 which are missing, because we come from an adjacent cell), and a test on the sign of e_2 for the 8 vertices determines if there can be at least one extremal point in the cell. At last, if this is possible,

we compute the extremal point on the segment of the extremal line that we are currently following, if it exists (i.e. the label associated to the end points of the segment are opposite), with linear interpolation.

The randomized implementation of the Marching Lines allows to extract the main extremal lines (i.e. the longest ones, which appeared experimentally to be the most reliable ones) of the 3D image, with only very few seeds (with respect to the total number of voxels), randomly distributed in the 3D images. The probability of missing an extremal line that way is approximately proportional to the inverse of its length. This method reduces drastically the number of computations to perform, compared with the extensive implementation. Even if the set of generated extremal points is not complete, it generally suffices to perform reliably the 3D registration.

We have implemented both methods (extensive and randomized), and we present in figure 12 the extremal lines extracted with the randomized version of the Marching Lines, from two 3D images of the same skull. Those segments are colored with the sign of second extremality e_2. The points where the color of the extremal lines is changed are the extremal points.

4 Geometric invariants associated with the extremal points

There are several geometric invariant values which can be associated with the points of the crest lines, and which can be used efficiently to perform the 3D registration, as it has been already shown in [6], for the case of the extremal lines registration. The geometric invariants are the values associated to the object which are invariant with respect to rigid transforms.

Mainly, for a point P, we can use the geometric invariants of the isosurface and the geometric invariants of the extremal lines at point P, plus geometric invariants corresponding to the relative position of the extremal lines with respect to the underlying surface.

For the surface (see figure 5), we can compute the trihedron $(\vec{n}, \vec{t_1}, \vec{t_2})$ of the principal directions at point P, and we have also the two principal curvatures k_1 and k_2. We can transform those two values into two other equivalent values, $K = k_1 k_2$ and $H = (k_1 + k_2)/2$; the Gaussian and mean

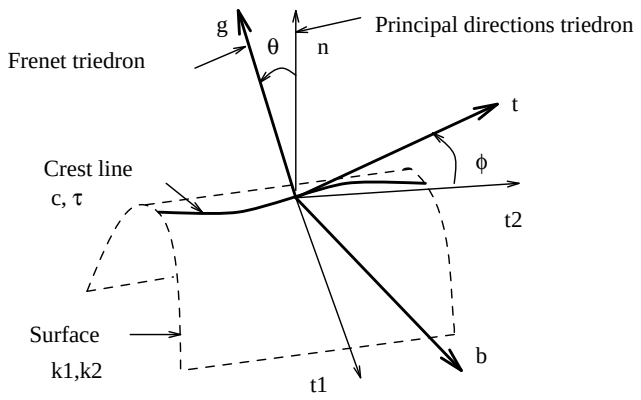

Figure 5: Invariants associated to crest line points

curvature of the surface, whose expressions are simpler than those of k_1 and k_2.

For the extremal line, we can compute the Frênet trihedron $(\vec{g}, \vec{t}, \vec{b})$, where $\vec{t}$ is the tangent, $\vec{g}$ is the normal, and $\vec{b}$ the bi-normal to the 3D curve. Those values can be computed with the approximation of the extremal line with a 3D spline, as in [6], or with the tensor product of the convolution of the 3D coordinates of the points with a filtering function (for example a Gaussian function), along the 3D curve. We can also compute the curvature c and torsion τ of the 3D curve at point P that way.

At last, we can compute two angles θ and ϕ between the two trihedrons $(\vec{n}, \vec{t_1}, \vec{t_2})$ and $(\vec{g}, \vec{t}, \vec{b})$, which are also geometric invariants. There are only two angles, because $\vec{t}$ is (should be) in the plane $(\vec{t_1}, \vec{t_2})$. We have $cos(\theta) = \vec{n} \cdot \vec{g}$, and $cos(\phi) = \vec{t} \cdot \vec{t_2}$. Other equivalent values can be computed from those ones, which are the normal and geodesic curvatures, and also the geodesic torsion of the 3D extremal line with respect to the surface (see for example [3]).

To summarize, for each extremal point P, we are able to compute K, H, c_1, τ_1, θ_1, ϕ_1, c_2, τ_2, θ_2, ϕ_2 which are geometric invariants characterizing this particular extremal point. We know that those invariants are not all independent (for example, c can be computed from H, K, θ, ϕ).

The stability of those values may differ, for example : the curvature c is more reliable than the torsion τ, because it takes into account lower order derivatives. We do not know yet how to classify the geometric invariants from the more stable one to the less stable one.

5 Registration of two sets of extremal points

We describe now only briefly the registration method that we use to perform the automatic registration (included here for completeness). In fact, the most difficult part is to extract stable and invariant 3D points, the extremal points, with stable and invariant attributes, the geometric invariants. Due to the high precision of the computation of those features, the registration part is simple, fast and reliable, and can be achieved in many different ways than the one exposed here.

The method that we use is a prediction-verification scheme, with an iterative improvement of the final transform based on a least square fit, relying on the quaternion representation of the rigid transform, a method close to the one proposed by Besl in [1].

5.1 Objective measurements

The results of our experiments, however, are objective measurements which are relatively independent from the registration method, as we explain now :

In the following, we consider that we have n 3D points for the two models M_1 and M_2 to match, with m attributes attached to each point. The goal is to find the 3D rigid transform T that "makes the maximum number of points of the two models correspond the most".

Hence there are two simultaneous maximization to perform : the number of points matched, and the quality of the match.

First, we have to define when two points are said to be matched. We call M_1' the result of the application of the transform T to M_1 (M_1' is therefore very close to M_2). Two points P_1 and P_2 of the models M_1 and M_2 are said to match when the distance between the transformed point $P_1' = TP_1$ and P_2 is less than a maximum distance d. The quality of the transform T is the distribution $n'(d)$ of the number of points which are matched, up to a maximum distance d. We can say that the matching is successful, when the

number of points matched is greater than a given percentage p of the original number of points, that is, when $n'(d) \geq p \cdot n$. The advantage of defining the quality of the transform with only a subset of the points is to allow the finding of transforms, even when the two 3D images overlap only partially.

Hence, our criterion of success depends on two arbitrary values : p, the percentage of point matched, and d, the maximum distance of matching. From now, we will suppose that those two values are given by the user, or have been set for a given class of images. We have verified experimentally that for our types of images (MRI and Cat-scan) the result is not too sensitive to those two values, because, when a valid transform is found, the number $n'(d)$ of points matched is large, even when the distance d is small (this is due to the high quality of the extracted features, i.e. the extremal points).

As we can see, we could have also an evaluation criterion of the transform based on the attributes associated to the points (curvatures, torsions, ...) : two points match when both their position after geometric transform *and* their geometric invariant attributes are similar. We think that the invariant attributes are less discriminant than the positions of the extremal points, we do not use them in the matching quality criterion. As we will see, we use them only to reduce the complexity of the registration method.

5.2 Description of the registration method

- The first step (prediction) is to look for triplets of points from the two models M_1 and M_2 which can be put into correspondence, up to the distance d (we use the attribute values to reduce the complexity of this part).

- The second step (verification) is to consider the associated 3D transform for each couple of triplets which can correspond, and to verify if the associated transform is valid.

- The last step is to improve the transform with a least square fit, in order to match more points .

The theoretical number of possible triplets is n^3, and we should have therefore n^6 verifications to perform, for an exhaustive search : we need therefore a way to reduce the number of possible solutions.

5.3 Complexity reduction

We detail now the properties that we use to reduce the complexity, without going into the details of the implementation. These properties comes from the invariance of k_1 and k_2, and from the relative invariance of $\vec{t_1}$ and $\vec{t_2}$ (in our implementation, we have not used the invariants of lines : Frênet trihedron, line curvature and torsions). In order to use efficiently those invariants, we have established statistics between the two sets of extremal points after matching (see figures 7).

For the example of the skull, the standard deviation of the position error is 0.42 voxel, the orientation error is about 7 degrees, and the curvature error is about 0.1 $voxel^{-1}$. We detail now how we have used those invariants :

- As we saw, there are 16 different types of extremal points. A point of one type can be put into correspondence with only a point of the same type from the other model. Hence an important reduction of the complexity (n^6 to $16(n/16)^6$, but still $o(n^6)$).

- k_1 and k_2 are geometric invariants. Two points having too unsimilar principal curvatures are not matched (which can be quantified with the standard deviation of the curvature of figure 7).

- Distance for pairs of points. There are extra invariants which can be used to state if two pairs of points (P_1, P_2) and (P'_1, P'_2) are compatible. $||P_1\vec{,}P_2||$ is a new invariant and as the standard deviation of the points position is known, a bound can be set for the difference between $||P_1\vec{,}P_2||$ and $||P'_1\vec{,}P'_2||$.

- Orientations for pairs. In the same way, the projections of $\vec{t_1}$ and $\vec{t_2}$ on the axis $P_1\vec{P}_2$ gives 4 more invariants, which are $\vec{t_1}(P_1) \cdot P_1\vec{P}_2$ and $\vec{t_2}(P_1) \cdot P_1\vec{P}_2$, and the same for P_2. $\vec{t_1}(P_1) \cdot \vec{t_1}(P_2)$ and $\vec{t_2}(P_1) \cdot \vec{t_2}(P_2)$ are two more invariants (but the six are not independent). The bounds are set from the standard deviation of the orientations.

- Distances for triplets. To state if two triplets are compatible, we have 3 invariant distances, which can be ordered from the smallest to the largest one if we want an invariant vector.

- Orientations for triplets. 3 points define a base in 3D, $\vec{t_1}$ and $\vec{t_2}$ become then true invariants, which gives 3 more invariants for each point (only 3 because $(\vec{n}, \vec{t_1}, \vec{t_2})$ are orthonormal vectors).

To summarize, we can compute 2 invariants for points, 8 invariants for pairs of points, and 18 invariants for triplets of points (bases).

With 18 invariants, with the measured standard deviations of the positions, directions and curvatures, and for a given triplet of points of M_1, the average number of compatible triplets in M_2 is very close to 1 ! As all those invariants can be computed and stored into hash tables, their retrieval is performed in constant time, which reduces the number of predicted associated bases to verify to less than $16(n/16)^3$. In fact, we compute first the compatible points, then the compatibles pairs of points, and only then the compatible triplets, which also reduces the number of predictions to verify.

At last, we use some heuristics in order to process the most interesting bases first (for example by considering the pairs and triplets with the farthest points first, and by processing points with higher curvature first). As we stop the search as soon as a given proportion of the points are matched, only a few predictions are verified before an acceptable solution is given (one or two in practice).

5.4 The verification part

This part is easier to explain. The previous step generates a set of possible transforms $\{T\}$. We build then a 3D hashing table with the 3D spatial coordinates of the points of model M_2, and with a bucket size of d.

We can consider each of those transforms T. Let's take one of them, and call it T^0, from now, only 3 points are matched (the two triangles). We apply T^0 to each point P_1 of M_1, which gives P_1', and we check in the hash table (in 8 adjacent buckets) if there exists a point P_2 in M_2 whose distance to P_1' is less than d. This gives the number of points matched $n'^0(d)$, i.e. the quality of this transform.

If $n'^0(d) > 3$, we compute a new transform T^1, using a least square fit and the quaternion representation of the transform, such as described in [1], and we compute also the new quality of the matching $n'^1(d)$.

We iteratively repeat this operation, as long as the number of points matched n' strictly increases, otherwise, we stop the process : we have then a

final transform T, and a final quality $n'(d)$. In practice, this iterative process stops at once when T is a "bad" transform, because of the very low density of extremal points (less than $1/1000^{th}$).

If $n'(d)$ is greater than $n \cdot p$, we say that transform is valid, and we stop the process, which reduces the computation time. Even if no transform has the required quality $n \cdot p$, we can keep track of the best transform found T (according to n'), and give it as a result. This shows that the whole method depends mainly on one single arbitrary value, d, the maximum distance between two matched points.

5.5 Statistical analyses

Developing an exact statistical analysis of our algorithm is very difficult, because it is very dependent on the feature points quality (the measured standard deviations, which are a-posteriori measurements), and because the number of parameters is important (up to 18 invariants for 3D bases). But it would be a shame not to use the associated invariants simply because we have no exact theoretical results about them.

We prefer therefore the global a-posteriori analysis of the registration algorithm performances, which is to measure the total number of generated hypotheses N_h, the number of hypotheses N_g which lead to good matches, and the hypothesis which gave the first good match (see figure 8).

5.6 Conclusion about the registration method

Our method applies because we have been able to extract a set of extremely reliable 3D points, with associated geometric invariant attributes, from the two images. For any other kind of landmarks, surfaces or 3D curves, the solution is much more ambiguous because the distance between the two models is hard to define and compute (surface to surface or 3D curve to 3D curve distances). In our case, we really have a point to point correspondence between the two models, because the extremal points are very sparse : in the skull experiments, there are about 3000 extremal points, extracted from a 5 million voxels image, which means an approximate "density" of $1/1500^{th}$ extremal points per voxel. About $1/6^{th}$ of the points are matched, thus there is no ambiguity about the validity of the solution, when one is found. This is

not the case for surface matching methods, which are subject to local minima pitfalls.

The distribution of points matched according to a given distance, $n(d)$, is an objective way to measure the quality of the registration, and, because the number of extremal points is small, the verification part is fast.

Furthermore, because we have also access to the geometric invariants and are able to use them to reduce the set of possible transforms, we can explore systematically all those transforms, with a complexity which is still acceptable.

6 Experimental results in 3D

We have tried our registration method successfully with couples of MR images and couples of XRay scanner images, of the same patient. So far, it has never failed to find a good transform. The only parameters that we give now to our registration software "EpiMatch", which are image dependent, are the iso-value thresholds, and the resolutions (the sizes of the voxel) of the two images to register (we have arbitrarily fixed the maximum distance d to 0.6 voxel in all our experiments).

6.1 Quantitative results

The first example is the registration of two XRay scanner images of a skull, of about $160 \times 200 \times 140$ pixels each, with a spatial resolution of $1 \times 1 \times 1.5mm$.

The extraction of the extremal points takes about the same time than the extraction of the crest lines, that is, about 15' for each 3D image, with a Dec 5000 workstation (the bottleneck is a pre-filtering of the image, which takes 10' each). The registration part takes about 30 seconds CPU-time.

The second example is the registration of two MR images of the brain. Figure 11 presents the two images after automatic registration. The resolution is $1 \times 1 \times 3mm$.

We display in figure 6 the statistics about the registration quality : $Voxels$ (total number of voxels), $Model1$ and $Model2$ (number of extremal points in each models) , $Matched$ (number of registered points). Figure 7 is the standard deviations between the matched points, after registration : error for the positions (in voxels), for the principal curvatures k_1 and k_2 (in $voxel^{-1}$), and

	Voxels	Model 1	Model 2	Matched	perc.
skull (CAT-scan)	5 million	2849	2928	551	19 %
brain (MRI)	3.5 million	1841	1625	233	14 %

Figure 6: Registration statistics

	position	k_1	k_2	$\vec{t_1}$	$\vec{t_2}$
skull (CAT-scan)	0.42	0.16	0.02	9.2	7.0
brain (MRI)	0.42	0.08	0.02	6.5	6.2

Figure 7: Measured standard deviations between matched points

for the principal directions $\vec{t_1}$ and $\vec{t_2}$, in degrees. This shows the remarkable robustness of the measurements.

Figure 8 are statistics about the registration process : $first$ (first acceptable hypothesis), N_h (total number of generated hypotheses), N_g (number of good hypotheses), $aver$ (average number of points matched for good hypotheses).

As we can see, the first good hypothesis is found at once (first or second hypothesis to be verified), and there is very few differences between the different acceptable transforms found ($aver$ is very close to $Match$).

The results are better with isotropic data (cubic voxels), with a higher proportion of points matched (up to 50 %). We give in figure 9 the repartition between the different types of extremal points, for the case of an isotropic skull (90^3 voxels). M and m are for local maxima and minima (of the absolute

	first	N_h	N_g	aver	Matched
skull (CAT-scan)	2	4049	3506	535	551
brain (MRI)	1	1320	953	225	233

Figure 8: Registration process statistics

	Model 1	Model 2	Matched	perc.	stand
M1M2	322	307	178	55 %	0.29
M1m2	433	419	188	43 %	0.29
m1M2	128	121	45	35 %	0.25
m1m2	250	256	100	39 %	0.30

Figure 9: Major types of extremal points

value). For example $M1m2$ is for largest curvature maximum and second curvature minimum. For each major type, we have summed up the statistics of the four sub-types depending on the signs of the curvatures (the results between sub-types are similar). As we can see, the more stable extremal points are maxima of both principal curvatures (55% matched).

6.2 Qualitative results

Figure 12 presents the set of crest lines extracted from the 2 CAT-scan images, where color changes indicate the position of the extremal points. Figure 13 presents the superimposition of the two sets of registered crest lines.

Let us see now to what extend this method is precise. Figure 14 presents the two registered lines sets viewed from the bottom, we can perfectly see the crest line of the *foramen occipitale* (the closed loop in the center), which is the place where the skull articulates with the vertebral spine. Image 15 is a zoom of the foramen occipitale, the perpendicular segments are one voxel long (1mm), which helps to perceive the quality of both the extracted feature points and of the registration method.

We have also pointed out a particular anatomic point, called the *Opisthion*, which has been distinguished as a very reliable anatomical landmark by the physicians, and which is automatically extracted with our purely geometric method.

Figure 11 presents the two automatically registered MR images of the brain (right, registered image). We can see in this figure a horizontal black line in the middle of the brain, which is an anatomical feature that exists in both registered images. This line can never be seen complete in any slice

of the second MR image before registration (figure 10, right), which shows how a slight perturbation of the z axis orientation of the images can induce very different aspects in the slices. This shows also the importance of the precision of the registration, when we are to compare two images of the same patient, taken at two different times, for example to evaluate the evolution of a pathology.

7 Perspectives

The present MRI case is extracted from a 800 3D MR images study of the brain, performed in the Brigham and Woman's Hospital (Boston), to quantify the evolution of the Multiple Sclerosis, and for which precise and automatic registration is a crucial step (see [11]).

Another important thing is that the extremal points are invariants of 3D surfaces, hence the scope of their applications is far larger than 3D image processing. For example, they are defined also for 2.5D depth map, and can be extracted more easely from them than from 3D images, because the surface map $z = f(x, y)$ is already parametrized, which makes the computation of e_1 and e_2 easier : the classical formulae of differential geometry are generally given for parametrized surfaces. We have experimented successfully the registration of synthetic depth map images, but we do not possess yet information about the stability of the extremal points for real data, in that case.

8 Conclusion

We have presented a new class of geometric landmarks of 3D surfaces, the extremal points. We have shown an algorithm to extract automatically those extremal points from 3D images. We have also described a registration algorithm, based on those points, which can be used to perform reliably the registration of two 3D images, and to compute the stability of both the extremal points locations, and their associated geometric invariant attributes. In the future, we hope to extend the use of extremal points to inter-patient registration and determine the subset of those points which are also anato-

mical points, that is, stable from patient to patient.

Acknowledgment

I want to thank Olivier Monga, André Gueziec, Serge Benayoun, Alexis Gourdon and Nicholas Ayache for stimulating discussions about differential geometry. Part of this study has been supported by the Esprit Basic Research Action VIVA. Thanks also to Digital Equipment Corp. who provides us with fast computers, GE-CGR of Buc, France, for the two scanner images of the head phantom. At last, thanks to Dr Ron Kikinis, from the Brigham and Woman's Hospital, Boston, who provided the MR scans.

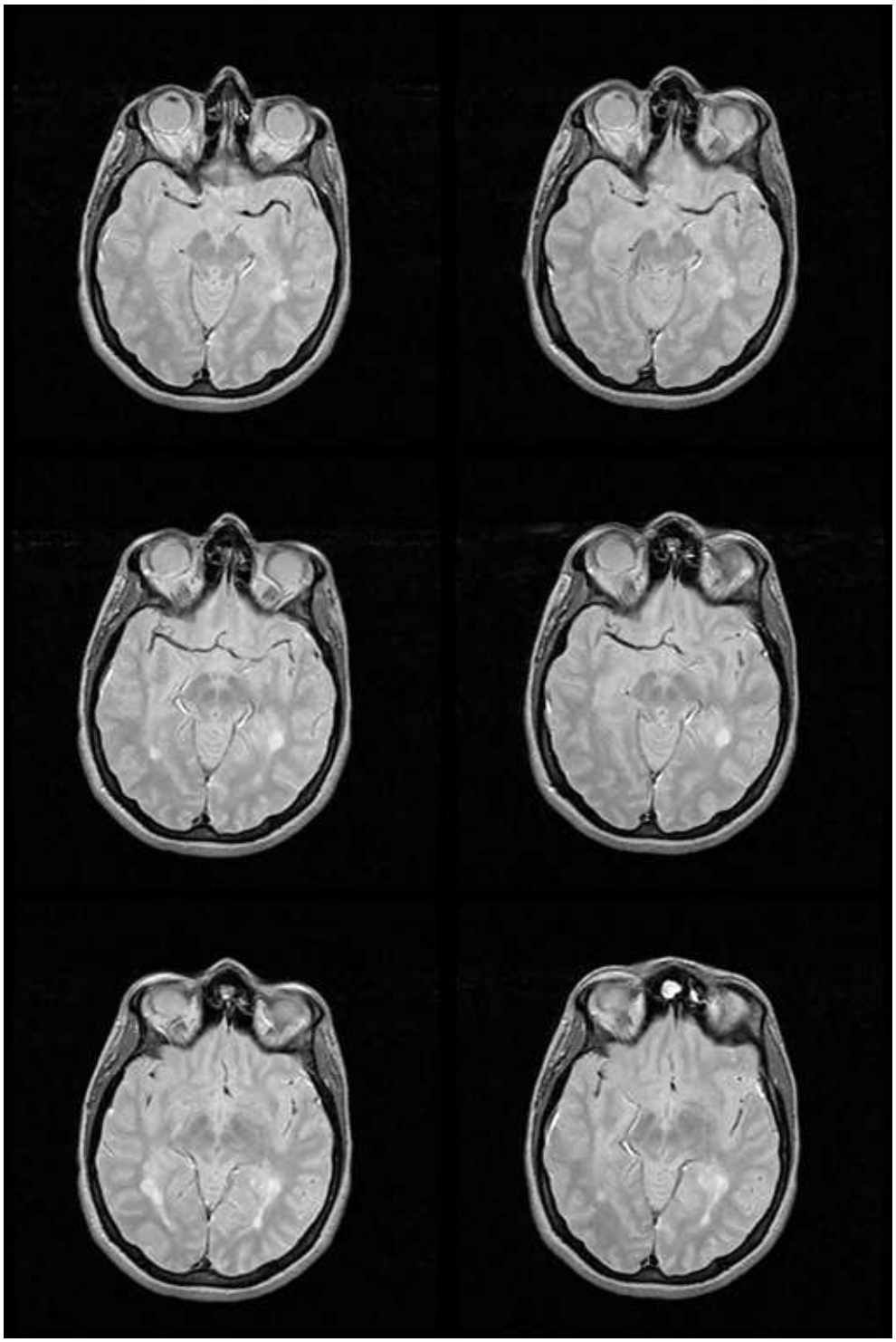

Figure 10: Two 3D images (left and right) of the same patient before registration. There is one month between the two acquisitions (1x1x3mm, 256x256x54 slices). 3 slices are displayed.

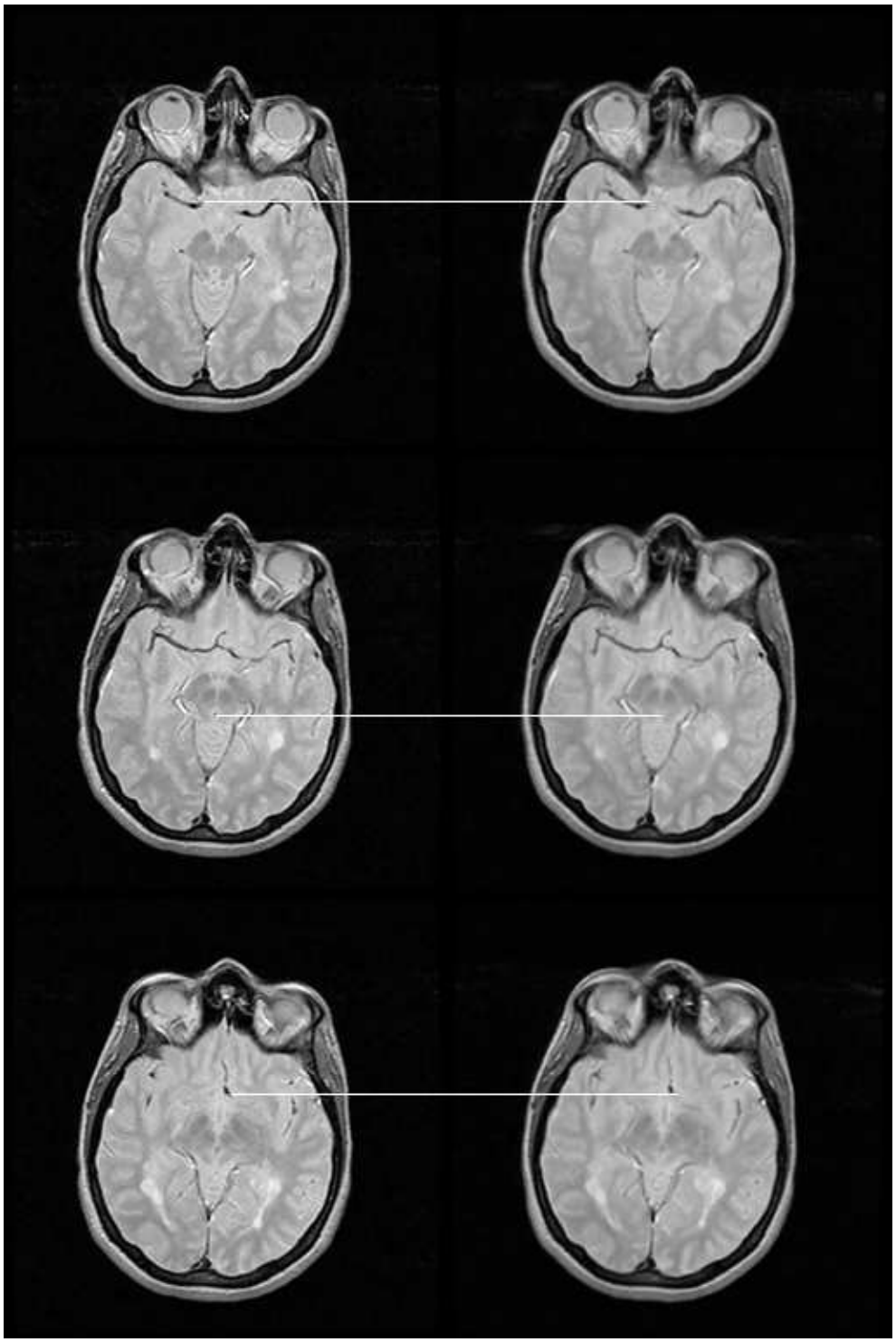

Figure 11: The two 3D images after automatic registration, with corresponding points (white lines). Right is the second image, resampled in the geometry of the first one (left). The horizontal black grove in the middle of the brain, which is visible in both registered images can never be seen complete in the other original image (previous figure, right) which shows the extreme importance of precise 3D registration for diagnosis.

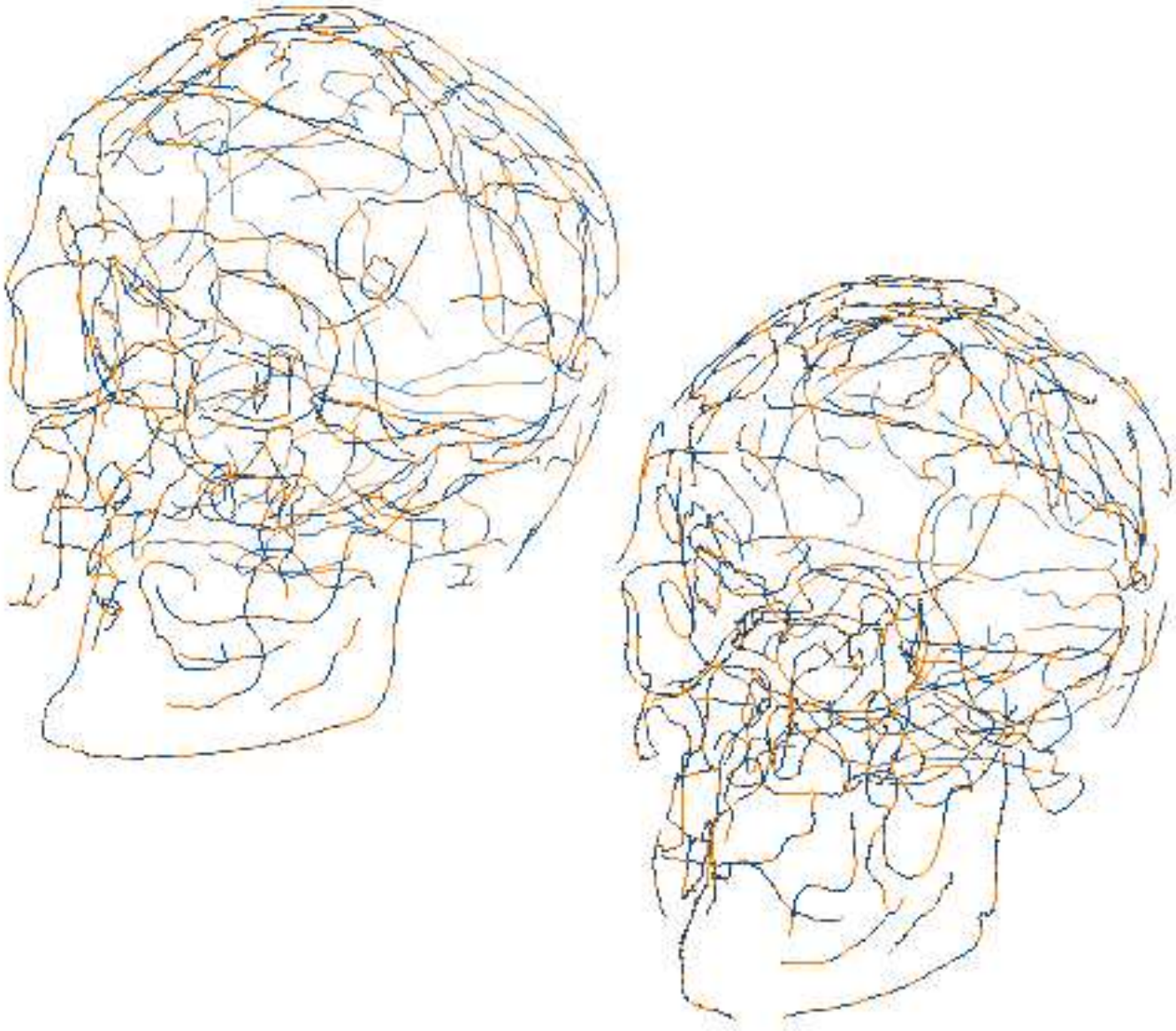

Figure 12: The two sets of crest lines, extracted from two different 3D Xray scanner images of the same skull (1x1x1.5mm) with the randomized version of the Marching Lines

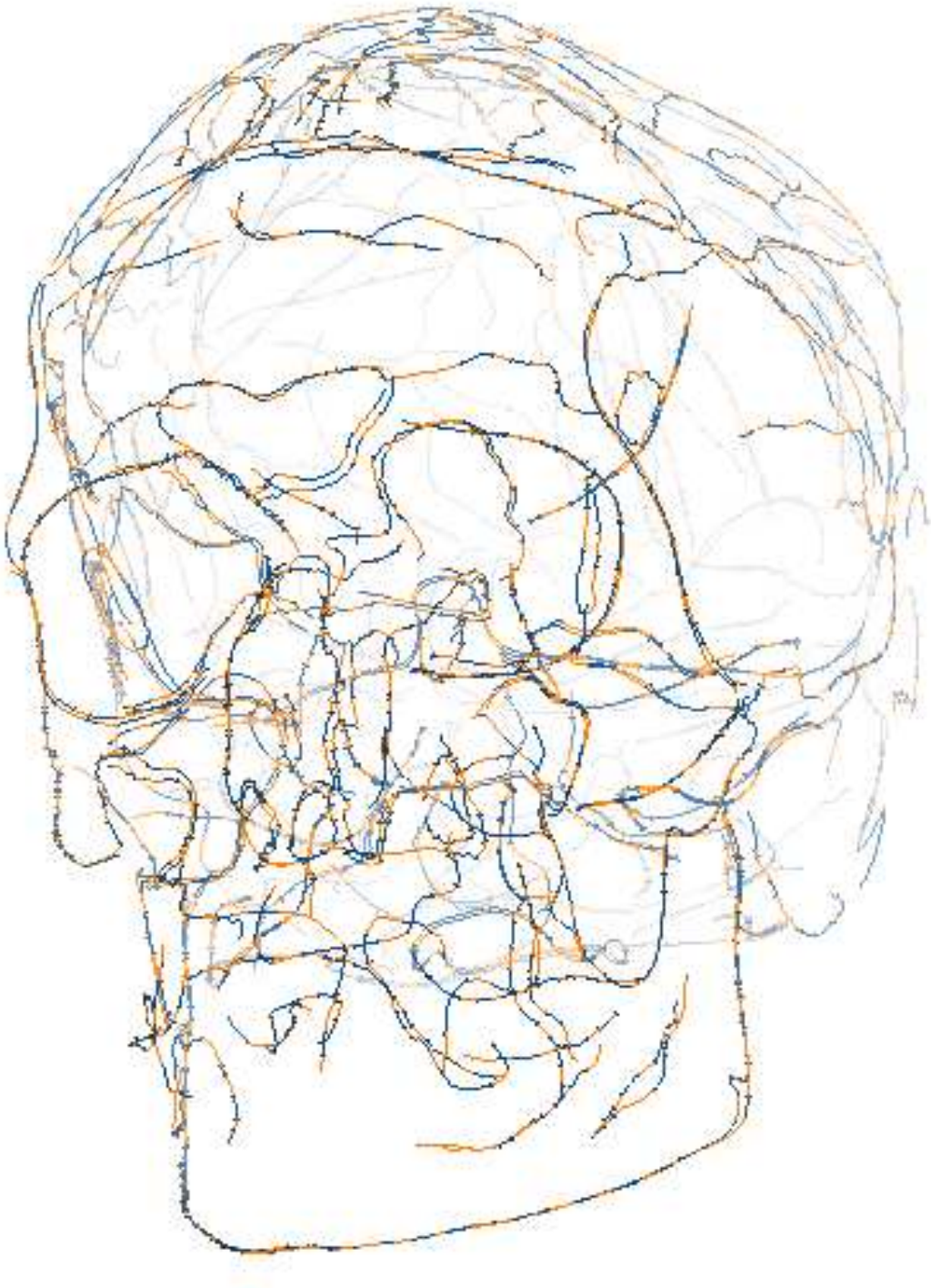

Figure 13: The two sets of lines after automatic registration, exactly superimposed

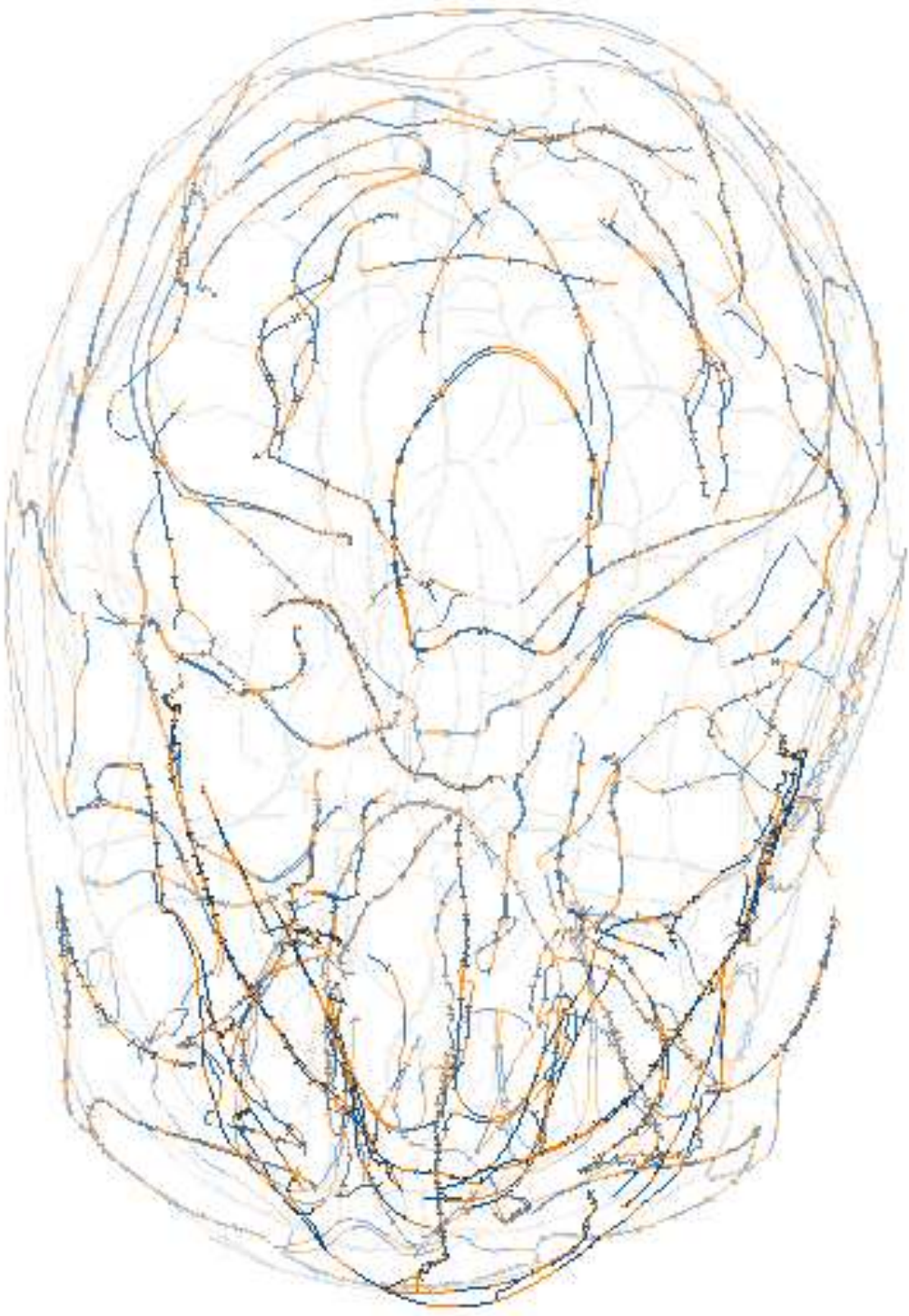

Figure 14: The two sets of lines after automatic registration, from the bottom, the closed loop in the middle is the foramen occipitale, where the spine and the skull articulate. The Opisthion point is emphasized.

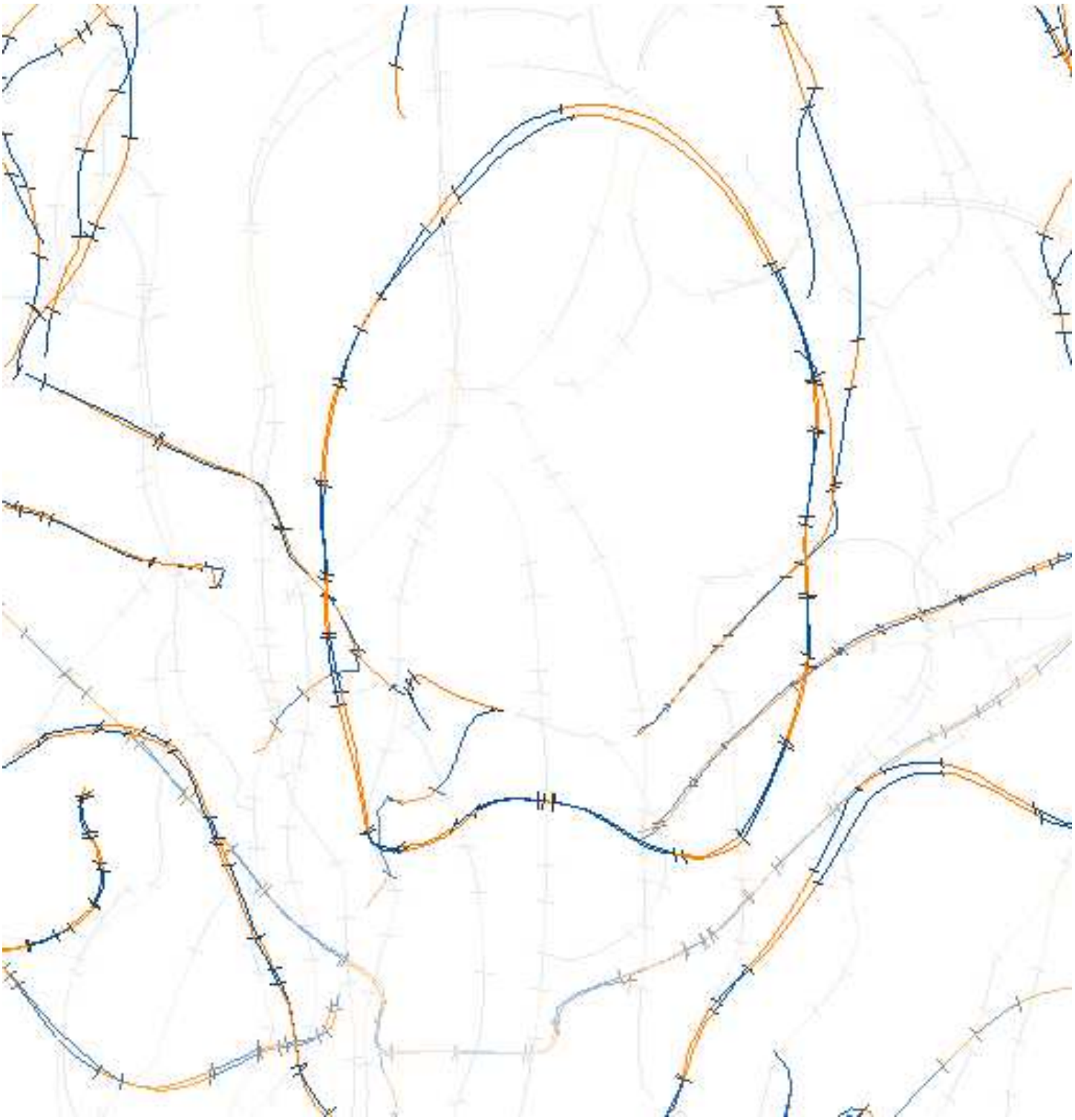

Figure 15: A zoom at the level of the Foramen Occipitale, to show the precision. The colors correspond to the sign of the second extremality. The black segments materialize the extremal points positions. The length of those segments is one voxel (1mm).

Appendix :

We recall here the equations, based on the use of the implicit functions theorem, which give the differential characteristics of the iso-intensity surface from the differentials of the image function f, up to order 2 (extracted from [16]). These equations are symmetric in the x,y,z coordinates.

- K,S, Gaussian and mean curvatures
- k_1,k_2 principal curvatures
- $\vec{t_1}$,$\vec{t_2}$ principal directions

$$\begin{array}{rl} K = \frac{1}{h^2}[& f_x^2(f_{yy}f_{zz} - f_{yz}^2) + \quad 2f_yf_z(f_{xz}f_{xy} - f_{xx}f_{yz}) + \\ & f_y^2(f_{xx}f_{zz} - f_{xz}^2) + \quad 2f_xf_z(f_{yz}f_{xy} - f_{yy}f_{xz}) + \\ & f_z^2(f_{xx}f_{yy} - f_{xy}^2) + \quad 2f_xf_y(f_{xz}f_{yz} - f_{zz}f_{xy})] \end{array} \tag{2}$$

$$\begin{array}{rl} S = \frac{1}{2h^{3/2}}[& f_x^2(f_{yy} + f_{zz}) - \quad 2f_yf_zf_{yz} + \\ & f_y^2(f_{xx} + f_{zz}) - \quad 2f_xf_zf_{xz} + \\ & f_z^2(f_{xx} + f_{yy}) - \quad 2f_xf_yf_{xy}] \end{array} \tag{3}$$

with :

$$h = f_x^2 + f_y^2 + f_z^2 \tag{4}$$

$$k_i = S \pm \sqrt{\Delta} \quad with \quad \Delta = S^2 - K. \tag{5}$$

$$\vec{t_i} = \vec{\alpha} \pm \sqrt{\Delta}\vec{\beta} \quad with \quad \vec{\beta} = (f_z - f_y, f_x - f_z, f_y - f_x) \tag{6}$$

We give the x coordinate of $\vec{\alpha}$, the y and z coordinates are obtained with circular permutation of x,y,z.

$$\begin{array}{rl} \vec{\alpha}\cdot\vec{x} = -\frac{1}{2h^{3/2}}[& -2f_z^3f_{xy} \quad +f_y^3f_{zz} \quad +2f_y^3f_{xz} \quad -2f_y^2f_zf_{xy} \\ & +2f_z^2f_xf_{yz} \quad +2f_z^2f_yf_{xz} \quad -2f_y^2f_xf_{yz} \quad -2f_zf_xf_yf_{zz} \\ & +2f_xf_yf_zf_{yy} \quad +f_y^2f_zf_{xx} \quad -2f_z^2f_xf_{xz} \quad +f_zf_x^2f_{zz} \\ & -f_x^2f_zf_{yy} \quad +2f_z^2f_yf_{yz} \quad -f_zf_y^2f_{zz} \quad +f_z^3f_{xx} \\ & -f_z^3f_{yy} \quad -2f_y^2f_xf_{xz} \quad +2f_x^2f_yf_{yz} \quad -f_y^3f_{xx} \\ & +2f_xf_z^2f_{xy} \quad -f_yf_z^2f_{xx} \quad -2f_zf_y^2f_{yz} \quad +f_yf_z^2f_{yy} \\ & -2f_zf_x^2f_{yz} \quad +2f_xf_y^2f_{xy} \quad +f_x^2f_yf_{zz} \quad -f_x^2f_yf_{yy}] \end{array} \tag{7}$$

References

[1] P. J. Besl and N.D. McKay. A method for the registration of 3d shapes. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 14(2):239–256, 1989.

[2] L. Brunie, S. Lavallée, and R. Szeliski. Using force field derived from 3d distance maps for infering the attitude of a 3d rigid object. *Proc. 2nd ECCV*, pages 670–675, 1992.

[3] Manfredo P. Do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice Hall, 1976.

[4] John M. Gauch. *Multiresolution Image Shape Description.* Springer-Verlag, 1992.

[5] Gerard Giraudon and Rachid Deriche. On corner and vertex detection. In *Conference on Computer Vision and Pattern Recognition*, Hawaii (USA), June 1991.

[6] A. Guéziec and N. Ayache. Smoothing and matching of 3D-space curves. In *Proceedings of the Second European Conference on Computer Vision 1992*, Santa Margherita Ligure, Italy, May 1992.

[7] H. Jiang, R.A. Robb, and K.S. Holton. A new approach to 3-d registration of multimodality medical images by surface matching. *Proc. SPIE Visualization in Biomedical Computing*, pages 192–213, 1992.

[8] Les Kitchen and Azriel Rosenfield. Gray-level corner detection. *Pattern Recognition Letters*, 1:95–102, 1982.

[9] Jan J. Koenderink. *Solid shape.* The MIT Press, 1990.

[10] G. Malandain and J.M. Rocchisani. Registration of 3d medical images using a mechanical based method. *Proc. 14th Int. Conf. IEEE EMBS, Sat. Symp. on 3D Advanced Image Processing in Medecine*, pages 91–95, 1992.

[11] D. Metcalf, R. Kikinis, C. Guttmann, L. Vaina, and F. Jolesz. 4d connected component labelling applied to quantitative analysis of ms lesion temporal development. *IEEE EMBS*, November 1992.

[12] Olivier Monga, Serge Benayoun, and Olivier D. Faugeras. Using partial derivatives of 3d images to extract typical surface features. In *Proceedings CVPR '92, Urbana Champaign, Illinois*. IEEE, July 1992. also an INRIA Research Report (1599).

[13] Alison J. Noble. Finding corners. *Image and Vision Computing*, 6:121–128, 1988.

[14] C.A. Pelizzari, G.T.Y. Chen, Spelbring D.R., R.R. Weichselbaum, and C.T. Chen. Accurate three-dimensional registration of ct, pet, and/or mr images of the brain. *J. Comp. Assist. Tomog.*, 13(1):20–26, 1989.

[15] J-P. Thirion, N. Ayache, O. Monga, and Gourdon A. Dispositif de traitement d'informations d'images tri-dimensionnelles avec extraction de lignes remarquables. Brevet Français, numero 92 03900, Mars 1992. Pattent pending.

[16] J-P. Thirion and A. Gourdon. The 3d marching lines algorithm and its application to crest lines extraction. *rapport de recherche INRIA*, (1672), May 1992.

[17] J-P. Thirion and A. Gourdon. The 3d marching lines algorithm : new results and proofs. *rapport de recherche INRIA*, (1881), March 1993.

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399