IEEE Computational Science & Engineering - Vol. 4, no. 4, Oct.-Dec. 1997, pp. 29-41
Special issue on Geometric Hashing.[a]

Medical Image Registration using Geometric Hashing

André P. Guéziec
IBM T.J. Watson Research Center

Xavier Pennec
MIT, A.I. Lab. and INRIA, Projet EPIDAURE

Nicholas Ayache
INRIA, Projet Epidaure

Abstract

We present recent work at INRIA on *registering* medical images using Geometric Hashing. Here, registering means determining the rigid displacement that aligns two images to obtain the best overlap of anatomical structures represented inside the images.

We analyze two methods that perform hashing on geometric invariants and use a voting procedure, inside a six dimensional accumulator implemented using a one dimensional hash table, to determine the best candidate rigid displacements. The first method uses mostly *unary* geometric invariants, with an option to use *binary* invariants, while the second method uses binary invariants exclusively. The first method uses characteristic *ridge* curves extracted on the surface of objects for registration purposes, while the second method uses characteristic *extremal* points.

We present registration results using both methods, and report experimental registration accuracy data. We also analyze the computational complexity of the registration and compare with the complexity of alternate registration methods. Using data from a joint study with an industrial partner we show that Geometric Hashing can provide a practical, fast, and accurate method for registering CAT scan images of the same patient.

1 Introduction

1.1 Registration of Three Dimensional Medical Images

This article develops methods for registering 3D medical images. By registering, we mean determining the rigid displacement (combined rotation and translation) that aligns two images to obtain the best overlap of anatomical structures from the two images.

Image registration is recognized as a fundamental problem in Medical Imaging. Fusing and comparing medical images is very useful, and sometimes necessary, for diagnosis, patient follow up, treatment planning and surgical planning. Today a great variety of medical image modalities are available, such as conventional X-ray, X-ray Fluoroscopy, X-ray Computed Tomography (CT), Magnetic Resonance Imagery (MRI), Positron Emission Tomography (PET), Ultrasound (resulting images are sometimes called Sonograms), Single Photon Emission Computed Tomography (SPECT). Without an accurate geometric superimposition, it is hazardous to compare or fuse images corresponding to the same anatomy.

The registration can be performed for a single patient, or across patients (for instance for building a digital anatomical atlas), using a single image modality or multiple modalities. The registration can be rigid, as studied in the present article, or non-rigid, whereby various deformations can be applied to images in order to superimpose them. In general terms, the techniques that have been proposed fall into the following two broad categories: (1) intensity or pixel (voxel) based techniques, and (2) feature based techniques. Intensity based techniques define a pixel similarity measure between different images ("mutual information", entropy, or simply cross-correlation) and attempt to optimize registration parameters in order to maximize this measure. There is a priori no need to *segment* the image data, i.e. extract regions, surfaces, curves or points of particular interest. Although the first intensity based algorithms were generally slow, ineffective at optimizing the registration parameters, and used an objective function of limited relevance, there has been much progress recently in these three respects. Feature based techniques rely on

Medical Image Registration

- L. Lemieux *et al.* A patient to computed tomography image registration method based on digitally reconstructed radiographs. *Medical Physics*, 21(11):1749–1760, November 1994.
- W.M. Wells *et al.* Multi-Modal Volume Registration by Maximization of Mutual Information. *Medical Image Analysis*, 1(1):35–51, 1996.
- C. Studholme *et al.* Registration measures for automated 3-D alignment of PET and intensity distorted MR images. In K.V. Mardia, C.A. Gill, and I.L. Dryden, editors, *Image Fusion And Shape Variability Techniques*, 16th Leeds Annual Statistical Workshop, pages 186–193, Leeds, UK, July 1996.
- F. Maes *et al.* Multimodality image registration by maximization of mutual information. *Transactions on Medical Imaging*, 16(2):187–198, April 1997.
- C.A. Pelizzari *et al.* Accurate 3-D registration of CT, PET, and-or MR images of the brain. *Journal of Computer Assisted Tomography*, 13(2):20–26, 1989.
- G. Malandain and J.M. Rocchisani. Registration of 3-D medical images using a mechanical based method. In *16th Annual Int. Conf. of the Engineering in Medicine and Biology Society (EMBS)*, Rennes, France, November 1992. IEEE.
- S.M. Pizer *et al.* Zoom invariant fusion of figural shape: The mathematics of cores. Technical Report 96-004, University of Northern Carolina at Chapel Hill, 1996.
- J. Feldmar *et al.* 3D-2D projective registration of free-form curves and surfaces. *Computer Vision and Image Understanding*, 65(3):403–424, 1997.
- A.P. Guéziec *et al.* Registration of computed tomography data to a surgical robot using fluoroscopy: A feasibility study. Technical Report 20561, IBM T.J. Watson Research Center, 1996. http://www.research.ibm.com:8080/PS/8341.ps.gz.
- P.A. van den Elsen *et al.* Medical image matching – A review with classification. *IEEE Engineering in Medicine and Biology*, 12(4):26–39, March 1993.

a pre-processing step to identify various anatomical structures in both images. Such structures are then registered, and the transformation that was found to best superimpose the structures is applied to the underlying images.

1.2 Our approach

The approach we developed in the Epidaure project at INRIA (France) consists of extracting characteristic curves, called "Crest Lines" on the surface of anatomical structures inside each image, and of registering such lines using Geometric Hashing. In section 2 we develop the method of Guéziec and Ayache for registering surfaces (and three dimensional images) using characteristic curves [1]. In section 3, we expose the method of Pennec and Ayache [2] that was developed later and builds upon the first method.

One of Guéziec and Ayache's original contributions was to exploit curve Frénet frames to directly compute a rotation and translation for each matching pair of curve sample points (instead of using three pairs of matching points as was traditionally done). Also, geometric invariants are computed on curve sample points, and are used for indexing the samples. Such invariants are *unary* they only need one sample point and its neighborhood to be computed. Instead, Pennec must specify a *basis* feature with respect to which *binary* invariants are computed. This increases the computational complexity of the method but makes it more robust. The methods also differ in the way they quantize the invariant space (the accumulator) and determine the contribution of a particular transformation to various accumulator cells. Both presentations are accompanied with experimental results and analyzes of computational complexity.

The main reason for preferring Geometric Hashing over other matching methods is its advantageous computational complexity. This is an important issue in Medical Imaging because of the size and intricacy of structures of interest embedded inside three dimensional images. In section 2.3 we study in detail the complexities of various matching methods for the particular problem of matching curves and motivate our choices: Geometric Hashing is more efficient in general and particularly efficient when one object covers a small portion of the object(s) in the database. Geometric Hashing is very well adapted when one wishes to register a shape against a database of shapes, as the complexity of the algorithm is largely unaffected by the size of the database. This capability is used in section 2, as individual curves can be recognized and labeled, as well as in section 3. We believe that the problem of registering against a database is likely to become important in future medical imaging applications using digital anatomical atlases or digital databases of diagnosis images, that are currently being developed as pilot projects.

2 Registering 3D Medical Images Using Geometric Hashing On Characteristic Curves

This section presents a method for registering 3D medical images using characteristic curves. Details of this method and various experimental results can be consulted in Ayache and Guéziec's publications [1, 3].

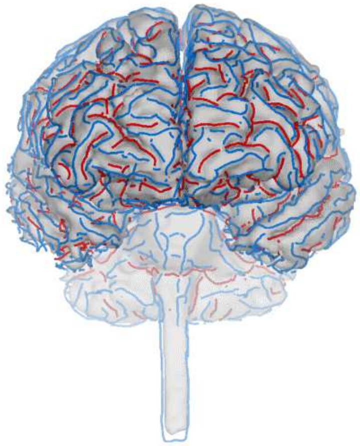

Figure 1: Example of crest lines on the brain surface (front view). Lines in red roughly correspond to sulci whereas blue lines could be interpreted as gyri (Courtesy of G. Subsol).

2.1 Extraction of Crest Lines

Crest Lines (also referred to as Ridges or Ridge Curves) are, according to our practical definition, locations of the surface where the maximum surface principal curvature in absolute value reaches a local maximum in the principal direction of maximum curvature. Definitions can vary slightly, but intuitively, such lines correspond to the surface's most salient features. Sample Crest Lines are shown in Fig. 1.

Prior to the late 1980s and early 1990s, the interest in Ridges (Crest Lines) was mostly theoretical, in areas of mathematics related to Catastrophe Theory. Practical applications were then discovered by researchers in Computer Vision, Graphics and Medical Imaging together with the specification of algorithms for extracting ridges. Several algorithms were used to extract Crest Lines from medical images. Fully automatic methods were developed by Monga *et al* and by Thirion and Gourdon. Cutting *et al* developed a method using a template of Ridges to assist in their extraction from image data. This method was extended by Dean *et al* (see side-bar).

The approach investigated by Guéziec was to build a differentiable representation of the surface using a B-spline surface: by decoupling surface u and v variables, by varying the surface resolution, by imposing constraints on the image gradient, and by using surface regularization parameters compatible with decoupling variables, Guéziec was able to build surfaces with sufficient detail and to generate meaningful Ridges.

Aside from the location of the Ridges, we can compute a series of intrinsic surface properties: principal curvatures and principal directions for a point that is not umbilic. Once Ridges have been extracted and their properties recorded, we can design a very efficient registration algorithm based on the idea of Geometric Hashing. Generally, there has been much more research on the registration of surfaces than on the registration of curves. Although the same techniques could potentially be applied, developing a special purpose algorithm for curves can result in increased efficiency and speed. This is what our method does.

Differential Geometry and definition of Crest Lines

- M.P. do Carmo. *Differential Geometry of Curves and Surface.* Prentice Hall, Englewood Cliffs, New Jersey, 1976.
- I. Porteous. *Geometric Differentiation: For the Intelligence of Curves and Surfaces.* Cambridge University Press, 1994.
- J.J. Koenderink. *Solid Shape.* M.I.T. Press, 1990.

Extraction of Crest Lines and Extremal Points

- O. Monga *et al.* Using third order derivatives to extract ridge lines in 3-D images. In *Proc. CVPR'92*, Urbana Champain (Il), June 1992.
- J-P. Thirion and A. Gourdon. The 3-D marching lines algorithm. *Graphical Models and Image Processing*, 58(6):503–509, 1996.
- C.B. Cutting. Applications of computer graphics to the evaluation and treatment of major craniofacial malformation. In Udupa and Herman, editors, *3-D Imaging in Medicine.* CRC Press, 1989.
- A. Guéziec. Surface representation with deformable splines: Using decoupled variables. *IEEE Computational Science and Engineering Magazine*, 2(1):69–80, 1995.
- J-P Thirion. New feature points based on geometric invariants for 3-D image registration. *Int. J. Computer Vision*, 18(2):121–137, 1996.

2.2 Description of the Curve Registration Algorithm

We are given a set of model curves $\{M_i\}$ and a scene curve S. We wish to identify a curve M_i which has the largest subset of points in common with S after a rigid transformation (matching); and specify that rigid transformation that minimizes a measure of the deviation between the points that were associated in the two curves (registration). Let m denote the number of model curves, and n the maximum number of sample points in a model or scene curve.

Figs. 2 and 3 illustrate our curve matching algorithm. In the preprocessing stage, we process each model curve in turn. For each sample point, we com-

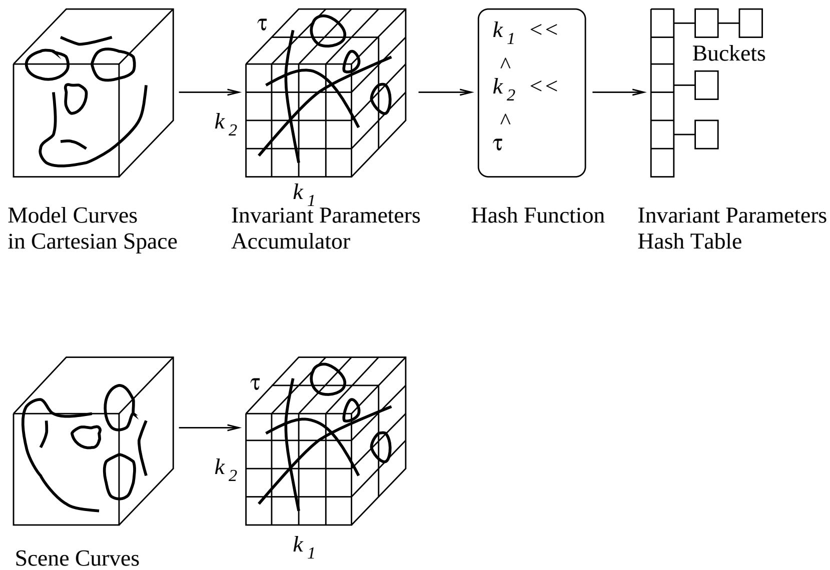

Figure 2: Preprocessing: after the computation of three intrinsic parameters (we have used five for the experiments of Section 2.5), each curve point falls in a cell of an imaginary three dimensional accumulator. The accumulator is implemented with a one dimensional hash table: the mapping from three dimensions to one is performed using a hash function. At recognition time, the same point, although in a different position and orientation, will fall in the same accumulator cell.

pute invariant parameters (discussion in section 2.4) and store the sample point inside an accumulator cell according to the parameter values. The principle of the method is the same independently of the number of parameters, or dimensionality of the accumulator. In Fig.2 three parameters are represented; this should not be construed as meaning that the number of parameters must always be equal to three. In most of our experiments, we have used five parameters.

At this point, it is important to note that accumulators are always implemented using a one-dimensional hash table with our approach. In a first pass, the parameter values are transformed to integers corresponding to a bin number for each dimension (quantization of the invariant space). Rather than constructing a high dimension table that would have space allocated for each potential cell, we use a hash function that maps a tuple of integers (bin numbers) to a single integer code. Typically the function uses a combination of binary "shift" and "xor" operations, to attempt covering the full range of the computer representation of an integer (role of the shift) and to distribute entries as much as possible (role of the xor). Such a hash function is represented with a pictograph using language "C" notation for the binary operators in Fig (2). An ideal hash function would distribute entries evenly over the entire integer range.

In the recognition stage, we process each sample point of each scene curve: we compute the set of chosen invariant parameters for each sample point and store (a pointer to) the point inside an accumulator cell corresponding to the parameter values. Next, pairs of points (M, S) from a scene and a model curves inside the same accumulator cell are tested as potentially corresponding points. A rotation and translation is then computed between the two points and associated frames and entered in a second accumulator, of dimension six. Specifically, if we are given the Frénet frames $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ and $(\mathbf{t}', \mathbf{n}', \mathbf{b}')$ corresponding to the points M and S, the rotation R which brings the two frames into correspondence is given simply by the outer product: $R = (\mathbf{t}', \mathbf{n}', \mathbf{b}')(\mathbf{t}, \mathbf{n}, \mathbf{b})^t$. The translational component is then easily obtained.

Although in the particular implementation that is discussed, the quantization of the six dimensional transformation accumulator is determined empirically, we can use statistics or methods for propagating uncertainty measurements to determine an optimal quantization. It is important to note that there is a notion of error zone in the vicinity of a transformation. The error zone is defined as an ellipsoid, whose six axes lengths correspond to uncertainty measurements. To avoid omitting votes because a particular transformation falls near the boundary of an accumulator cell, if the error zone of that transformation intersects other cells, then a vote is added to

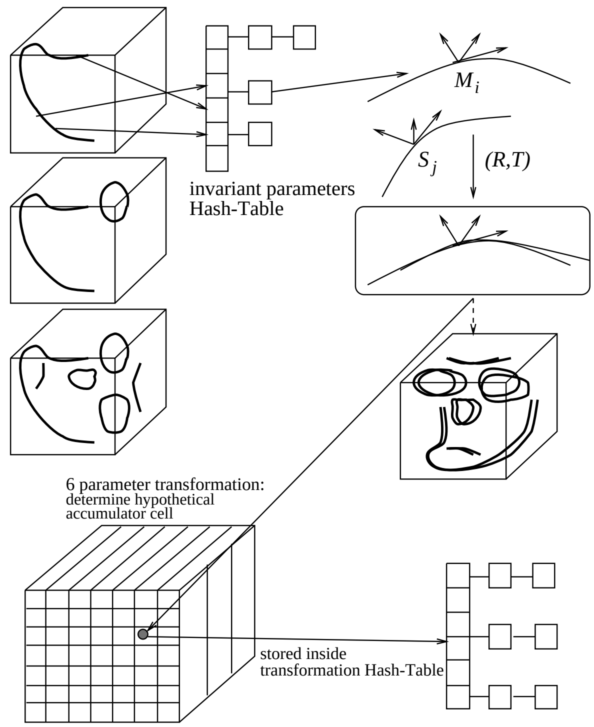

Figure 3: Recognition: each time a point of the scene curves is matched with a model point in the accumulator, the Frénet frames allow to compute the six parameters of a rigid displacement. This displacement falls in an imaginary six dimensional accumulator, which is represented using a second hash table. Buckets of the hash table corresponding to the most populated cells provide hypotheses for the registration parameters.

such cells as well. In this implementation, the error zone is the same for each transformation. In the implementation of section 3, the size and shape of uncertainty zones varies with the transformation. For practical purposes, once uncertainty values are determined, the cell width is chosen to be a small multiple of the uncertainty, say three times the uncertainty. This guarantees that a vote is added to at most 2^6 cells for each transformation. It is a good idea to enforce a bound on the number of cells affected by a given transformation: because of the dimensionality (6) of the accumulator, very large hash tables could be required.

Densely populated cells of this second accumulator are detected as hash table buckets that receive a high number of votes. For each such cell, we compute a refined rotation and translation, by retrieving the pairs of points that hash to the cell and by determining a new transformation that best registers such points. This can be done by solving a standard least squares problem, or by using robust least squares, or better by using a Kalman filter. The refined transformation that registers the largest number of points for a pre-specified maximum mean squared error is retained. For better stability, it is possible to combine the Geometric Hashing method with methods based on finding closest points. This is essentially what is done in the verification step of the algorithm described in the second part of this article.

The complexity of the recognition stage is between $O(n)$ and $O(mn^2)$, depending on the level of quantization of the parameter space. The length of a bucket's entry parameter space is at the worst $O(mn)$; the method depends on the existence of parameter variations along the model. The complexity of the pre-processing phase is $O(mn)$. However, this complexity number should be amortized because the hash table that was built can be re-used for a number of registration queries.

2.3 Comparison of Approaches to Curve Model Matching

The following approaches have been proposed for registering curves: Schwartz and Sharir's method [4] comprises a linear search step for the best offset between portions of curves. They also provide an innovative and powerful method for curve smoothing with bounded error. Ayache and Faugeras [5] hypothesize and then verify matching hypotheses. Their prediction-verification (or alignment) method is reviewed next. Zhang [6] identifies the closest pairs of points between two curves, computes a transformation using such pairs, and recomputes closest point pairs once the transformation has been applied. His method is similar to Besl and McKay's Iterative Closest Point method [7].

Our method is related to the method of Kishon *et al* [8] who compute the curvature and torsion of the curves and use such parameters for hashing. They use a polygonal representation of the curves, and thus vote for a model and a displacement length, representing a difference between the arc-length locations of the point s_l and the candidate matching point $m_{i,j}$ measured relative to some reference point along each curve's representation. Since our representation of the curves includes a differentiable structure and thus Frénet frames, we may explicitly compute the rigid transformation as part of the matching process.

We now study the complexity of four methods for curve matching. Our study is related to the comparative study reported in Lamdan and Wolfson [9], with the following differences: each method, except

Complexity / Method	Alignment	Accumulation	Geometric Hashing	ICP
lower bound	$O(mln)$	$O(mn \log n)$	$O(n)$	$O(mn \log n)$
upper bound	$O(mln^2)$	$O(mn^2)$	$O(mn^2)$	$O(imn^2)$

Table 1: Lower and upper complexity bounds for the four different methods of curve recognition: Alignment, Accumulation, Geometric Hashing and ICP.

the Iterative Closest Point method, uses *unary* invariant parameter values to select pairs of corresponding points, and computes a rigid transformation for each pair of points using Frénet frames.

In the Alignment method, l significant points of the curve S are selected. In the *prediction* stage, we match each point with a model M_i and a point along M_i having a similar set of invariant parameters. Each such matching constitutes a hypothesis for which we can compute a rigid transformation. This transformation is used to superimpose the model onto the scene and *verify* the number of correspondences. One can stop as soon as a sufficiently good hypothesis is found or explore all hypotheses and keep the best ones. The computational complexity of the prediction stage is between $O(ml \log n)$ and $O(mln)$, depending upon the level of discrimination of the invariant parameters. The complexity of the verification stage is $O(n)$. Since the number of hypotheses generated is between $O(ml)$ and $O(mln)$, the overall complexity is between $O(mln)$ and $O(mln^2)$.

In the Accumulation Method, for each point on the curve S and every point on a model M_i having similar invariant parameters, we compute the rigid transformation that registers them using the corresponding Frénet frames. We register a vote in a quantized six-dimensional rigid transformation accumulator. We then locate the buckets in the accumulator receiving a lot of votes. For each model, the complexity is between $O(n \log n)$ and $O(n^2)$, depending on the degree of discrimination introduced by the invariant parameters. The difference with Geometric Hashing is that a binary search is used in parameter space as opposed to hashing into buckets.

In the Iterative Closest Point method (ICP), for each n points on the scene curve, the closest point on a particular model is determined in $O(\log n)$ time in average and in $O(n)$ in the worst case. Once closest points are determined, a transformation is computed in $O(n)$ that registers the corresponding points. The entire operation is then repeated a suitable number of times i. The complexity is between $O(mn \log n)$ and $O(imn^2)$. It is also proportional to the number of initial positions.

The complexity results are summarized in table 1. In conclusion, the Geometric Hashing method offers computational advantages, especially in terms of its sub-linear complexity growth in the number of models.

2.4 Choice of Invariant Parameters

We first considered curvature and torsion as only invariant parameters. Curvature and torsion measured on a smooth curve are invariant to rigid displacements. Then the method was extended to semi-differential invariants, combining distance and angles measured with respect to a reference point. Statistical studies indicated that such semi-differential invariants could be more stable than curvature and torsion alone. This approach was later exploited by Pajdla and van Gool [10].

A few points on the curves called *basis points* must be selected and used to compute the parameters of the other curve points. For instance, we can compute angles, a Euclidean distance or an arc-length measure between any given curve point and a basis point. Choosing such basis points can be difficult; an interesting possibility is to use maxima of curvature. Also, since the parameters that are used for recognition involve pairs of points, the complexity of the algorithm is increased, and is in principle $O(n^2)$ or higher, where n is the number of points in the curves.

Exploiting the property that our curves are embedded on surfaces, we discovered additional intrinsic parameters, different than curvature and torsion. The algorithm extracting Ridges computes the surface normal $\mathbf{N}$, the maximum and minimum surface principal curvatures k_1 and k_2 as well as the associated directions $\mathbf{e_1}$ and $\mathbf{e_2}$ (or their opposite). For each point of a curve that we consider, two frames are available (Fig.4), (1) a surface frame $(\mathbf{e_1}, \mathbf{e_2}, \mathbf{N})$ and a curve Frénet frame $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ where $\mathbf{t}$ is the curve tangent, $\mathbf{n}$ the curve normal and $\mathbf{b}$ the binormal. A review of these notions can be found in a good textbook on Differential Geometry. We expect that $\mathbf{n}$ and $\mathbf{N}$ will be different as well as $\mathbf{t}$ and $\mathbf{e_1}$. Otherwise, the ridge would be a line of curvature, which is not expected in general.

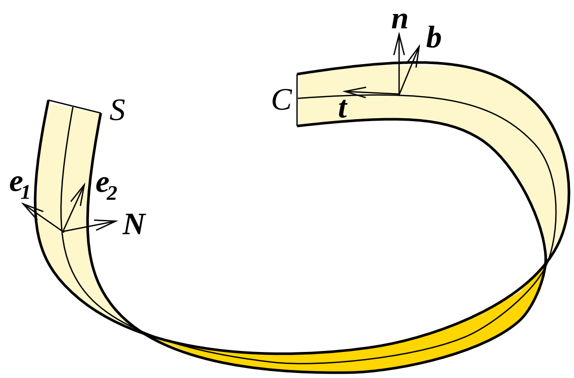

Figure 4: Surface ribbon centered around a space curve. For each curve point we can compute the Frénet frame $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ where $\mathbf{t}$ is the curve tangent, $\mathbf{n}$ the curve normal and $\mathbf{b}$ the binormal and the local surface frame $(\mathbf{e_1}, \mathbf{e_2}, \mathbf{N})$, where $\mathbf{e_1}$ and $\mathbf{e_2}$ are the principal directions of curvature in the surface tangent plane and $\mathbf{N}$ is the surface normal.

Our assumption is that we can choose curve sample points that are different from surface umbilic points, where $\mathbf{e_1}$ and $\mathbf{e_2}$ would not be defined, and where k_1 and k_2 would be equal. Various angles and parameters can be computed using both frames, the curve curvature and torsion and the surface principal curvatures. Not all the parameters that can be extracted are independent. We have identified a few combinations of five parameters. One of such combinations of parameters is the following: (1) curvature of the curve k, (2) torsion of the curve τ, (3) *geodesic torsion* of the curve with respect to the surface τ_g, and (4,5) two angles measured between the surface frame and the curve Frénet frame:

$$\left\{ \begin{array}{ll} \theta & = (\widehat{\mathbf{n}, \mathbf{N}}) \\ \phi & = (\widehat{\mathbf{t}, \mathbf{e_1}}) \end{array} \right.$$

The two invariants k and τ_g can be substituted with the principal curvatures k_1 and k_2 using the relations:

$$\left\{ \begin{array}{ll} k \cos\theta & = k_1 \cos^2\phi + k_2 \sin^2\phi \\ \tau_g & = (k_2 - k_1) \cos\phi \sin\phi \end{array} \right.$$

thus providing other combinations of invariants.

This raises the interesting question of whether the invariant parameters that we use are indeed independent: we observe that several surfaces can be locally different and have locally the exact same ridge (excluding obvious global differences). A very simple example is given by an ellipsoid E_1 of axes lengths $(a > b > c > 0)$ and an ellipsoid E_2 of axes lengths $(a > b > c/2 > 0)$. The longest (and sharpest) ridge will be precisely the same for E_1 and E_2 but the underlying ellipsoids will have quite different principal curvatures. Armed with this observation, we argue that in general the surface frame $(\mathbf{e_1}, \mathbf{e_2}, \mathbf{N})$ and the ridge Frénet frame $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ should not be related, and that the curvature and torsion of the curve should not explain the surface curvature entirely. Consider a surface ridge on an arbitrary smooth surface: while keeping the ridge in its exact same position, by analogy with the ellipsoid example, we can start "pinching" the surface and changing the local curvature and principal directions for various locations on the ridge. Since the curvature and torsion of the curve alone cannot explain the local surface geometry, angles relating the surface and curve frames must be specified, as well as the portion of the surface curvature not explained by the curve curvature, providing five independent invariant parameters.

In all cases, the curve Frénet frame must be computed independently from the algorithm that extracts ridges, by differentiating the curve. In order to compute the differential parameters of the curve, we build a differentiable representation using B-splines. This is a standard procedure for which we have added a few improvements. In particular, we have constrained the curve to be embedded in the surface by minimizing the sum of square products between curve tangent and surface normal. An additional benefit of this method is to apply a certain amount of smoothing to the parameters k_1, k_2 and τ_g as well as to the Ridge position, thereby removing some of the noise. The details of the curve fitting method can be consulted in [1].

2.5 Experimental Results

We consider two CT scan images of a skull phantom at a resolution of 256 by 256. The skull phantom was scanned in two different positions and orientations. Such images were processed by the algorithm of Thirion and Gourdon, resulting in the the sets of ridges displayed in Fig.5.a. The algorithm described in this section was then used to register the lines (Fig.5.b).

We performed registration experiments using two CT images of a vertebra scanned in two different positions and orientations. The result of ridge extraction is shown in Fig.6.a. The result of registering the ridges using our algorithm is shown in Fig.6.b.

Within the framework of a joint study with G.E. Medical Systems France, our goal was to assess registration accuracy for the particular application of single modality, single patient registration. This partic-

Figure 5: **(a) Left:** Crest lines on two 3D CT images of a skull obtained with the algorithm of Thirion and Gourdon. **(b) Right:** Superimposition of the lines of **a.** with our software in 7.7 seconds CPU on a DEC 5900. 8 curves are matched out of 15 and 32 curves. 53 points are matched out of 666 and 1124. Note that these experiments were performed before the DEC alpha architecture was available. CPU timings would be significantly decreased today.

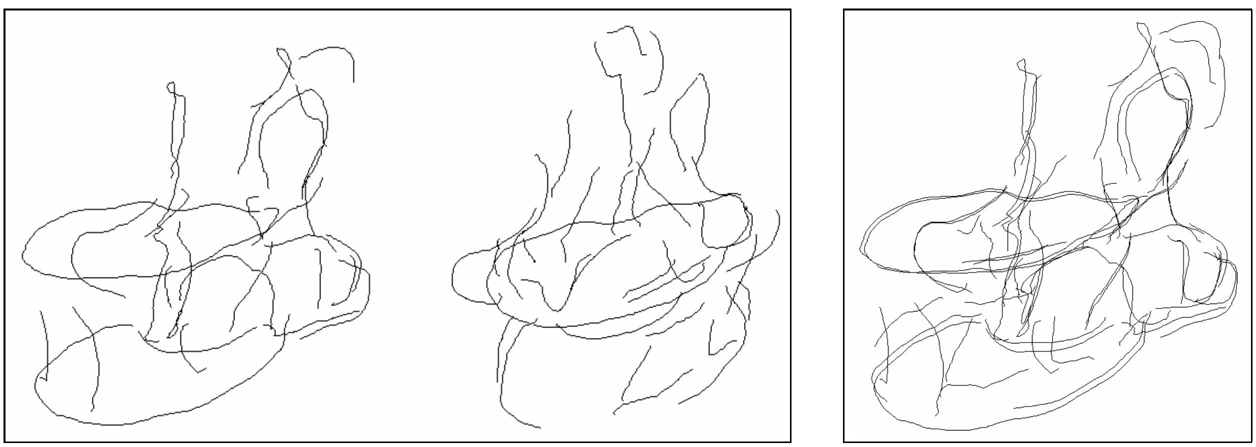

Figure 6: **(a) Left:** Crest lines on two 3D CT images of a vertebra. **(b) Right:** Superimposition of the lines of **a.** with our software in 8.9 seconds CPU on a DEC 5900. 12 curves were identified out of 28 and 33. 82 points were identified among these curves out of 1250 and 1213 points respectively in the two different sets of curves.

ular problem has several important real world applications. This technology applies for post-operative or post-treatment assessment and patient follow-up studies. Another potential application consists of fusing data-sets acquired in different orientations, to compensate for the anisotropy of the process of building 3D volume data by stacking scanned 2D slices.

We used a calibration phantom specifically designed for the study. The phantom was built using a skull phantom and an affixed pair of "N" shaped tubular markers. The phantom was imaged (CT scanned) in two different positions and orientations (256 x 256 x 30 images with 1mm slices). The second position was registered to the first position using our method of ridge curve matching. We then computed the distance between points sampled on the "N" shaped markers in the first view and corresponding points on the second view after registration. Such distances were found to be between 0.05 mm at the "N" shaped marker centers and 0.85 mm at the marker peripheries. Most of the areas of clinical interest were situated in the vicinity of the "N" shaped marker centers, indicating that the 0.05 mm registration error would more likely apply. The registration technology that was described in the present article was transferred to the company Aleph Medical in Grenoble (currently: Focus Medical).

3 Substructure Matching with Geometric Hashing on Frame Features

In the previous section, we have first modeled objects (anatomical structures) using surfaces and then using curves embedded on the surfaces. In order to simplify the matching and registration process, it is often necessary to go one step further and extract isolated landmarks. Indeed, if we could select on each line a mean number of l features that are supposed to be well conserved, we could focus on these l features only instead of the n points of the whole line. The complexity of the previous algorithm would then be reduced from a lower bound of $O(n)$ to $O(l)$. Moreover, isolated features, or landmarks, could appear even in the absence of curves, as with the modeling of proteins in section 3.2.

However, the increase in the number of lines from the previous section and the increase of the noise with real patient images drastically augments the density of invariants in the hash table. This could lead to an important number of false positives that overwhelm the correct matches. The maximum complexity would then be reached and the algorithm could even provide a wrong answer.

In order to have a correct algorithm, we show in this section how to keep a highly selective hash table by using binary invariants instead of just unary ones. This approach is related to the semi-differential invariants methods previously discussed. The drawback is an increase of the complexity from $O(n)$ to $O(l^2)$ (in practice, we observe that the lower bound is respected and the correct solution is always found).

3.1 Registering Medical Images with Frames

To obtain precisely located features, Thirion introduced new feature points called extremal points, which are points of locally maximal curvature in both principal directions of the surface. These points are specific points on the crest lines and can be seen as the generalization of corner points for smooth surfaces in 3D. Another way to view such points is to consider extremal lines instead of just crest lines. Extremal lines are defined as the loci on the surface where at least one of the principal curvatures is extremum in the associated principal direction. Extremal points are at the intersection of these lines.

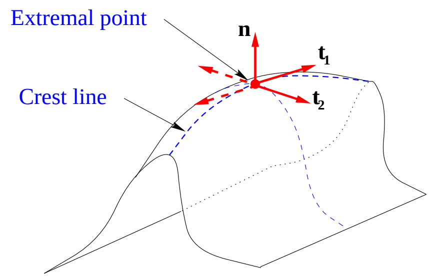

Figure 7: Crest line with an extremal point, modeled using a frame.

To the position of the point, we can add the normal vector $\mathbf{N}$ and the two principal directions $\mathbf{e_1}$ and $\mathbf{e_2}$ of the surface to constitute a local coordinate system or a frame (Fig. 7). In this context, each medical image is modeled by a set of frames and the matching problem is to find correspondences between a subset of these frame sets in different images. We have developed in [11] methods to solve the dual problem of registration of matched frames (i.e. finding the rigid transformation).

Preprocessing Step To obtain an invariant representation with respect to the global position and orientation of the considered structure, we can express the configuration of all frames relative to one frame (the basis). For efficiency, we store this representation in a hash table and, for correctness, we include the uncertainty of each invariant. As only part of the frames are in the same configuration in the two images, the one chosen as the basis may not be present in the other image. The preprocessing step is thus repeated with each frame as the basis.

Recognition Step We choose a frame of the second structure (the scene) to be the basis for computing the invariant representation and retrieve, thanks to the hash table, what are the compatible model frame couples. If the basis belongs to a common substructure, then a significant number of frames are in the same configuration with respect to it. We then match the model and scene bases (Fig. 8).

This process is repeated for every amino acid as the basis to find its possible matches in the model and we only keep the matches that are above a given threshold (typically 4 or 5 in our applications).

Clustering Compatible Matches and Verification. For each individual match, we maintain dur-

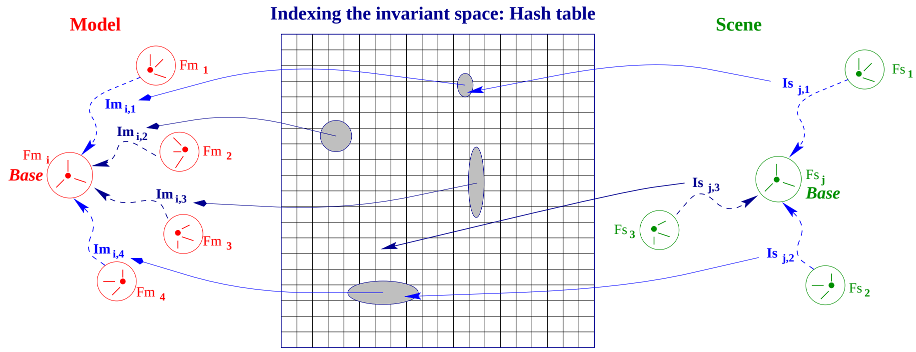

Figure 8: **Preprocessing:** the 6D invariant vector associated with every couple of model frames is computed with its error zone and used as an index for the couple in the hash table. **Recognition:** for each scene frame couple, we compute the 6D invariant vector and retrieve through the hash table every compatible model frame couple. For each such couple, we tally a vote for the matching of the reference frames (here the match (Fm_i, Fs_j) scores 2).

ing the recognition step an estimation of the associated transformation by fusing the transformations between confirming frames. To group matches belonging to the same (rigid) substructure, we run a very simple clustering algorithm on the associated transformation. Indeed, up to measurement errors, frames should undergo a similar transformation within a single substructure. Each cluster is then refined by an iterative closest neighbor technique where we enforce symmetry of the matches and verify their validity with a χ^2 test.

In the case of medical images, matching model and scene crest lines is not sufficient since different line matches can correspond to different transformations. Thus, we run once again our clustering algorithm on transformations to find out the compatible line matches and we obtain a single transformation from the model to the scene image.

Registration of CT images of the skull We present in Fig. 9 an example of the registration of two CT images of the skull of the same patient acquired in different positions. About 75 crest lines are matched with more than 4 extremal points among the 550 in each image leading to a total of 550 matched extremal points (only on the 75 matched lines). Using the techniques described in [12], we have computed that the registration accuracy (the expected standard RMS on image super-imposition in the area of matched features) is 0.04 mm.

Registration of MR images of the head Fig.10 is an example of registration of two MR images of the same patient. In this case, 80 crest lines are matched among the (about) 620 in each image, for a total of 350 matched extremal points. The accuracy of the registration is 0.06 mm. In this case, we have 24 images of the same patient, which allow us to run the statistical validation method we developed in [12]. It confirms our accuracy estimation.

3.2 Application to Molecular Biology

The recognition algorithm presented is this section was originally developed to find common substructures between proteins [2]. We show very briefly how this problem can be put in the same framework and show an application example.

Non-related proteins can have very different structures, but when they share a common biological function, often they share a common 3D substructure that realizes this function: this is a motif. Binding motifs can be widely distributed in the protein sequence and involve only few, if any, consecutive amino acids. Thus we cannot really use the sequence and we have to focus on the pure 3D structure. Moreover, the nature of amino acids themselves can vary since only part of the residues are used to bind. The only meaningful information is thus the geometric configuration of the amino acids in space. This observation leads Fischer *et al* [13] to apply the geometric hashing technique to this problem.

Each amino acid has 4 atoms participating to the backbone of the protein, 3 of them always being in the same geometric configuration. We use these 3 atoms to define the configuration of the amino acid in space (Fig. 12). Each amino acid is thus modeled as a frame (as were extremal points in medical images), and a protein is modeled by an unordered set of frames (as were crest lines before). To find similar substructures

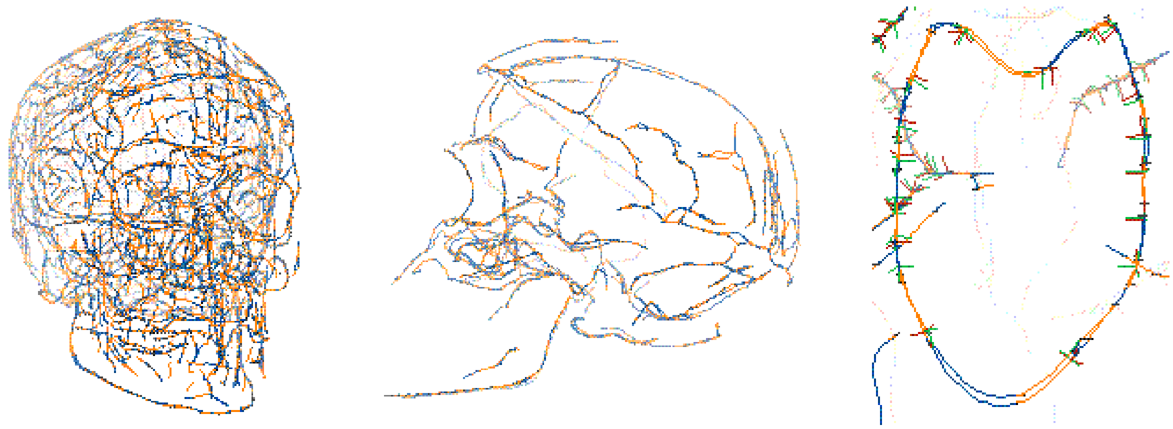

Figure 9: Example of registered crest lines between two CT skull images of the same patient. Extremal points are represented by a color change from yellow to blue on the lines. **(a) Left:** Front view with all crest lines from the two skulls. **(b) Middle:** Left view of the matched crest lines. **(c) Right:** Closeup on the occipital foramen on the right. In this last image, the width of a line is a tenth of a voxel, which shows the very precise registration of these extremal points. One can also see that the trihedron part of the matched frames is very well conserved.

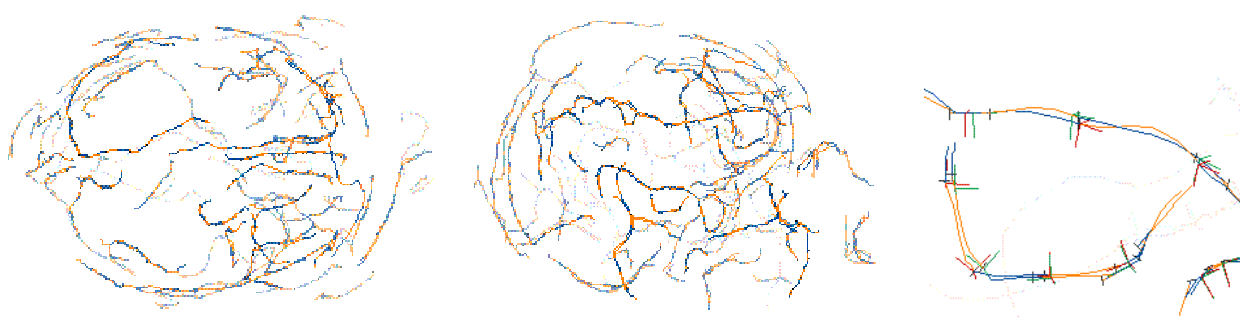

Figure 10: Example of registered crest lines between two MR images of the brain (same patient). Extremal points are represented by a color change from yellow to blue on the lines. **(a) Left:** View of matched crest lines from the top. **(b) Middle:** view from the right of the head. **(c) Right:** closeup to show the quality of the frame matches.

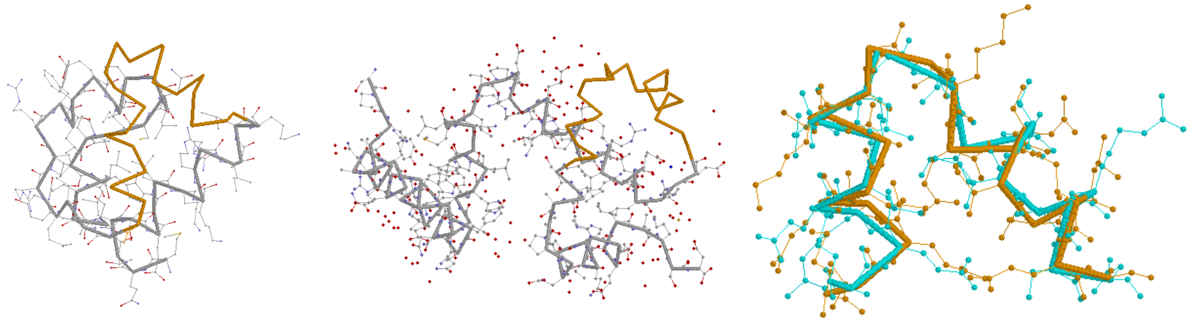

Figure 11: **(a) Left:** The CRO protein (2CRO) of phage 434. **(b) Middle:** The tryptophan repressor of E. Coli (2WRP). The matched part is displayed in yellow. **(c) Right:** Registered matches found between 2CRO and 2WRP: the HTH motif. We can see that not only the backbone is very well matched, but also collateral chains are pretty well conserved.

in two proteins, we now have to find two subsets of frames that are in the same configuration, up to a global rigid transformation. Since this is exactly the problem we address previously for medical images, we can use exactly the same algorithm.

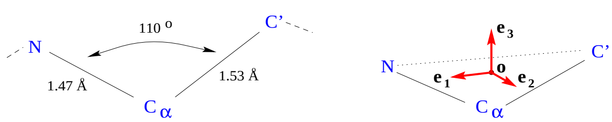

Figure 12: Modeling an amino acid by a frame.

We next study two proteins known to bind to the DNA in two different organisms: the tryptophan repressor of E. Coli and CRO protein of phage 434. These two proteins bind tightly to the DNA at a gene's promoter, thus preventing RNA polymerase from fixing and hence blocking the initiation of the transcription (they are repressors). They should thus have a common substructure that realizes this biological function.

The comparison of the two previous proteins gives one principal common substructure which is, as expected, the Helix-Turn-Helix motif, responsible for the binding to the DNA. The registration of the HTH motifs in the two proteins shows that not only the backbone is very well matched, but collateral chains are also well conserved. The use of frames instead of just points allows the algorithm to discard several false matches of amino acids (4 in the HTH case). Moreover, it makes the algorithm robust with respect to parameters that tune the quality of the matches (uncertainty attached to frames on input).

4 Discussion

In the first part, we have build upon previous methods for registering curves to incorporate geometric invariants associated to a ribbon, or strip, of surface. Following the indexing step of Geometric Hashing in the space of invariants, we have introduced a six dimensional accumulator to efficiently select the rigid transformation that register the maximum number of curve points. Our algorithm thus acts as a combination of two filtering procedures for isolating the best candidate transformations. The computational complexity is somewhere between $O(n)$ and $O(n^2)$, where n is the number of sample points chosen on the curves to be registered. Experimental results using data from a joint study with a Medical Imaging manufacturer indicate that the method is suitable for registering same-patient CAT scans because of its repeatability, speed, and accuracy. In the second part, we have developed a new extension of geometric hashing based on frame features. We show that Geometric Hashing can be successfully applied to very different problems such as substructure matching of proteins and matching of medical images.

The examples shown in this article indicate that Geometric Hashing can provide a powerful method for registering highly detailed three dimensional medical images of the same modality (CT registered against CT or MR against MR). The levels of registration accuracy that can be obtained are compatible with clinical requirements; also, the computational performance is very good, which allows to perform numerous registration attempts and favors experimentation. Aside from registration, Geometric Hashing also works for recognition: we believe that Geometric Hashing is a promising method for registering against a database of images, and for automatically recognizing a feature, such as a specific anatomic structure, or the absence thereof, inside an image. Also, Geometric Hashing provides useful statistics on the number of features that were paired and on how well they could be registered; this information, which is unbiased by a user-specified initialization, can be used for evaluating the registration.

Our experience with the method has taught us a few limitations: a primary requirement is to isolate corresponding geometric features: low level of detail imaging modalities, such as SPECT and PET lack geometric features that Geometric Hashing uses; generally, it is difficult to extract reproducible features for multi-modality registration; for the case of deformable registration, the difficulty of finding invariant parameters hampers the method. As discussed in Section 3, when there are few corresponding features, or when there is significant noise corrupting the features, the method can break down unless serious uncertainty handling techniques are implemented.

5 Acknowledgements

We wish to thank J. Knoplioch of General Electric Medical Systems Europe, for providing data sets used in studying the accuracy of registration, as well as J.L. Coatrieux, for providing CT images of a vertebra.

References

[1] A. Guéziec and N. Ayache. Smoothing and matching of 3-D space curves. *International Journal of Computer Vision*, 12(1):79–104, January 1994.

[2] X. Pennec and N. Ayache. An $\mathcal{O}(n^2)$ algorithm for 3D substructure matching of proteins. In *Proceedings of the first International Workshop on Shape and Pattern Matching in Computational Biology*, pages 25–40, June 1994. also as INRIA Research Report n^o 2274.

[3] N. Ayache, A. Guéziec, J.P. Thirion, and A. Gourdon. Evaluating 3-D registration of CT-scan images using crest lines. In *Mathematical Methods in Medical Imaging II*, volume 2035-06, pages 60–71, San Diego, July 1993. SPIE.

[4] J.T. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *International Journal of Robotics Research*, 6(2):29–44, Summer 1987.

[5] N. Ayache and O.D. Faugeras. HYPER: A new Approach for the Recognition and Positioning of Two-Dimensional Objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):44–54, 1986.

[6] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.

[7] P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[8] E. Kishon, T. Hastie, and H. Wolfson. 3-D curve matching using splines. In *First European Conference on Computer Vision (ECCV)*, Antibes, France, April 1990. Springer Verlag.

[9] Y. Lamdan and H. Wolfson. Geometric Hashing: A general and efficient model based recognition scheme. In *International Conference on Computer Vision*, pages 238–249, 1988.

[10] T. Pajdla and L. van Gool. Matching of 3-D curves using semi-differential invariants. In *International Conference of Computer Vision*, pages 390–395. IEEE, 1995.

[11] X. Pennec. Registration of uncertain geometric features: Estimating the pose and its accuracy. In *Proc of the First Image Registration Workshop, November 20-21 1997, Greenbelt, Maryland, USA*. CESDIS, 1997.

[12] X. Pennec and J.P. Thirion. A framework for uncertainty and validation of 3D registration methods based on points and frames. *Int. Journal of Computer Vision*, 25(3), 1997.

[13] D. Fischer, O. Bachar, R. Nussinov, and H Wolfson. An efficient automated computer vision based technique for detection of three dimensionnal structural motifs in proteins. *J. of Biomolecular Structures and Dynamics*, 9(4):769–789, 1992.

Andre Gueziec *is a Research Staff Member in the Visual and Geometric Computing Department at the IBM T.J. Watson Research Center. His research contributions are in medical imaging, scientific visualization and geometric modeling. His interests also include computer graphics and robotics.*

Gueziec earned his Ph.D. in computer science from University Paris XI in 1993. He graduated from the Ecole Centrale Paris in 1989. From 1989 to 1993 he was PhD student at University Paris XI, while doing research at INRIA in Paris. In 1993-1994 he was with the Courant Institute, New York University. He is member of IEEE and ACM. Readers can contact Gueziec at the IBM T.J Watson Research Center, Yorktown Heights, N.Y 10598.
E-mail gueziec@watson.ibm.com.

Xavier Pennec *graduated from the Ecole Polytechnique of Paris (1992) and received his Ph.D. from the same institution (1996) while working in the Epidaure group at INRIA (French Research Institute on Computer Science and Automatic Control), Sophia-Antipolis (France). He is currently a postdoctoral associate at MIT, Artificial Intelligence Lab.*

Dr Pennec's research interests include geometric data processing, geometry-based object recognition, geometric uncertainty and statistics, medical image analysis and protein structure analysis.
E-mail, Xavier.Pennec@sophia.inria.fr; WWW,
http://www.inria.fr/epidaure/personnel/pennec/pennec.html

Nicholas Ayache *is a Research Director at INRIA Sophia-Antipolis, France, where he leads the EPIDAURE research group on medical image analysis and robotics since 1989. He is also teaching graduate courses on computer vision at Univ. Paris XI, Ecole Centrale and consulting for a number of private companies, including currently Matra-Cap-Systemes and Focus Medical.*

He received his Ph.D. in 1983, and his "Thèse d'Etat" in 1988 both in computer science from the University of Paris XI, on topics related to model based object recognition, passive stereovision and multisensor fusion. His current research interests are in medical image processing and analysis, shape and motion representation, rigid and non-rigid registration, tracking and analysis of deformable objects, image guided and simulated therapy.
E-mail: Nicholas.Ayache@sophia.inria.fr.