Smooting and matching of 3-D space curves

André Gueziec, Nicholas Ayache

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1544

Programme 4
Robotique, Image et Vision

SMOOTHING AND MATCHING OF 3-D SPACE CURVES

André GUÉZIEC
Nicholas AYACHE

Octobre 1991

RR_1544

Smoothing and Matching of 3-D Space Curves *

André Guéziec Nicholas Ayache

October 17, 1991

INRIA
BP 105
78153 Le Chesnay Cédex
e-mail: gueziec and ayache@bora.inria.fr

Programme 4
Robotique, Image et Vision

*This work was financed in part by a grant from Digital Equipement Corporation.

Abstract : We present a new approach to the problem of matching 3D curves. The approach has a low algorithmic complexity in the number of models, and can operate in the presence of noise and partial occlusions.

Our method builds upon the seminal work of [KHW89], where curves are first smoothed using B-splines, with matching based on hashing using curvature and torsion measures. However, we introduce two enhancements:

- We make use of non-uniform B-spline approximations, which permits us to better retain information at high curvature locations. The spline approximations are controlled (i.e., regularized) by making use of normal vectors to the surface in 3-D on which the curves lie, and by an explicit minimization of a bending energy. These measures allow a more accurate estimation of position, curvature, torsion and Frénet frames along the curve;
- The computational complexity of the recognition process is independant of the number of models and is considerably decreased with explicit use of the Frénet frame for hypotheses generation. As opposed to previous approaches, the method better copes with partial occlusion. Moreover, following a statistical study of the curvature and torsion covariances, we optimize the hash table discretization and discover improved invariants for recognition, different than the torsion measure. Finally, knowledge of invariant uncertainties is used to compute an optimal global transformation using an extended Kalman filter.

We present experimental results using synthetic data and also using characteristic curves extracted from 3D medical images.

Lissage et Reconnaissance de Courbes Gauches Bruitées

Résumé : *Nous présentons une solution originale au problème de la reconnaissance et du recalage d'une courbe gauche discrète. La spécificité du problème est la nécessité de conserver une faible complexité algorithmique en présence d'un très grand nombre de modèles, d'être robuste au bruit et aux occultations partielles.*

Notre approche est une continuation logique des travaux de [KHW89] fondés sur un lissage des points par une courbe régulière puis par une reconnaissance à l'aide d'une table d'indexation mais présente deux innovations importantes :

- *Pour une détermination plus fiable du modèle et de ses dérivées, les points discrets sont lissés par des splines en utilisant un critère d'erreur mixte et une distribution anisotrope de nœuds fondée sur la courbure locale et une régularisation exploitant la connaissance de la normale à la surface sur laquelle la courbe est inscrite et minimisant explicitement la variation de la courbure;*
- *la méthode de reconnaissance, à partir d' une table d'indexation de la courbure et torsion des courbes modèles, et dont la complexité est indépendante du nombre de modèles, utilise explicitement le trièdre de Frénet pour rendre la méthode véritablement insensible aux occultations partielles, ce qui n'était pas le cas des approches précédentes. Par ailleurs, une étude statistique de l'incertitude sur l'estimation de la courbure et de la torsion, permet d'ajuster de manière optimale la discrétisation de la table d'indexation, de préconiser l'emploi de nouveaux invariants plus stables et d'estimer de manière optimale les paramètres du recalage global à l'aide d'un filtre de Kalman étendu.*

Nous présentons des résultats sur des données synthétiques et sur des courbes caractéristiques extraites d'images médicales.

1 Introduction

Physicians are frequently confronted with the very practical problem of registrating 3D medical images. For example, when two images provided by complementary imaging modalities must be compared, (such as X-ray Scanner, Magnetic resonance Imaging, Nuclear Medicine, Ultrasound Images), or when two images of the same type but acquired at different times and/or in different positions must be superimposed.

A methodology exploited by researchers in the Epidaure Project at Inria, Paris, consists of extracting first highly structured descriptions from 3D images, and then using those descriptions for matching [ABB+89, ABC+90]. The structured descriptions usually come from the extraction of regions of interest, obtained with a 3D edge extraction [MDR91, HA92] algorithm, potentially in conjunction with curve or surface deformable models [CC90, CCA91]. Then, characteristic curves on the surface are extracted. These curves describe either topological singularities such as surface borders, borders of holes, and simple or multiple junctions, etc., (see [MBA91]), or differential structures, such as ridges, parabolic lines, and umbilic points [MAS91].

The characteristic curves are stable with respect to rigid tranformations, and can tolerate partial occlusion due to their local nature. They are typically extracted as a connected set of discrete voxels, which provides a much more compact description than the original 3D images (involving a few hundreds of points compared to several million). Fig. 1 shows an example of ridges extracted from the surface of a skull [MBF92]. These curves can be used to serve as an invariant of the skull and to establish landmarks to match skulls between different individuals, yielding a standard approach for complex skull modeling [Cut89].

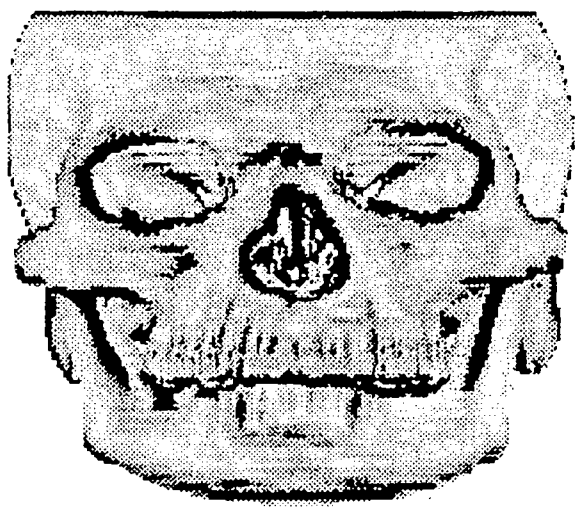

Figure 1: Extraction of characteristic curves (ridges) from the surface of a skull (using X-ray scanner image data)

The problem we address in this paper is the use of these curves to identify and accurately locate 3D objects. Our approach consists in introducing a new algorithm to approximate a discrete curve by a sufficiently smooth continuous one (a spline) in order to compute intrin-

sic differential features of second and third order (curvature and torsion). The analytical description of this curve allows the computation of the Frénet frame attached to each point of the curve (refer to [dC76] for a precise definition).

Given two curves, we then wish to find, through a matching algorithm, the longest common portion, up to a rigid transformation. We describe three possible approaches, specifically: prediction-verification, accumulation and geometric hashing. Using the third approach, which we call an indexation method, we introduce logical extensions of the work of [KHW89, SMM90, BBB87] in order to use splines to smooth and match points along curves in the presence of noise. Our work is also closely related to the work in [AF86] and [GLP84] on the identification and positionning of 3D objects.

We begin by outlining the problems and the approaches.

In Section 2, we discuss approaches to fitting curves to collections of voxels (points) in 3D imagery. Fig. 2 illustrates the problem of smoothing noisy data. The method that we ultimately use has two stages, and makes use of an adaptive criterion for smoothing. The two stages are summarized as follows.

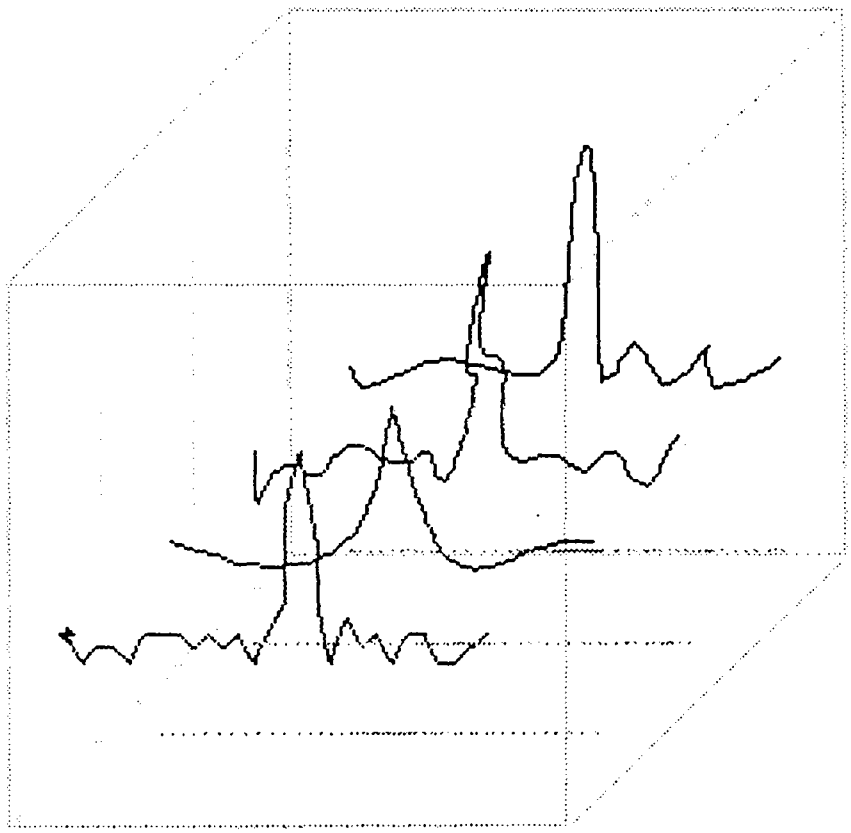

Figure 2: Given a noisy curve, the degree of smoothing should depend on the noise level, and the desired accuracy in the computation of curvature. All of these curves can be viewed as different noisy versions of a single curve.

1. We set a threshold ϵ_∞ which is a maximum allowed distance between the approxima-

tion and the curve. This threshold corresponds to the important notion of scale of observation. It is also linked to the noise level and to the spatial resolution of the 3D image. We then compute a polygonal approximation (of class C^0) of the curve using this threshold [DH73]. Using the vertices of this polygonal curve to guide the distribution of the knots, we construct a fourth-order B-Spline approximation of the initial data (the spline will be of class C^3).

The idea is to concentrate the knots in the neighborhood of high curvature points, which roughly corresponds to the results of a polygonal approximation [Pav77]. As a further refinement, it is possible to optimize the position of the control points by minimizing a mean square distance, ϵ_2, between points and the spline approximation.

2. The spline approximation is refined by making use of surface normals from knowledge of the surface on which the curve lies. Typically, the surface has been previously extracted as a noisy iso-intensity surface, by means of some 3-D (surface) edge detector. Knowledge of the surface normals constrains the approximating spline curve, and a penalization term is added to the quadratic criterion to be minimized that is used to define the spline approximation. The resulting spline thus balances a measure of the violation of the surface constraints against a measure of the bending or torsion energy based on a sum of squares of derivatives along the curve.

In Section 3, we discuss three classes of methods for rigid-transformation curve matching. By considering the algorithmic complexities, we then implement a matching system based on the indexation table (geometric hashing) approach, whose complexity is relatively independent of the number of models in the database. Certain modifications are required for use with the differentiable spline curve representation, and other enhancements are suggested, in order to make the method robust to partial occlusion of the curves (potentially in multiple sections). In sum, we considerably extend previous indexation-based curve-matching methods.

We further conduct a statistical study of various invariants that may be used to replace the curvature and torsion methods that we have used in our indexation-based matching, and suggest alternative invariants that might be used for hashing.

In Section 4, we provide experimental results obtained using synthetic and real data, and we indicate future work that is planned.

2 Approximation of noisy curves

2.1 Motivation for modeling with B-splines

Our goal is to begin with noisy data, given as a sequence of points in three-space, and to convert this data into a collection of evaluates of curvature, torsion, and Frénet frame measurements along the curve. A standard problem with such approximations is that a criterion for fit frequently allows large but local errors to be compensated by precision in tracking elsewhere. Further, the quality of the approximation should be independent of the resulting parameterization of the curve. Accordingly, we constrain the approximation to fit

the data to within a maximum deviation distance, which is a parameter that depends on knowledge of expected errors due to image acquisition, discretisation and boundary detection (see [MAS91]). Especially large deviations must be dealt with as outliers, and filtered before forming the representation.

One possible approach involves the use of the polygonal curve formed by joining the points. This solution clearly satisfies the maximum deviation criterion, but also tracks the noise, supplying no smoothing. Another method, used by Schwartz and Sharir [SS87] is to form the shortest polygonal path through a tube surrounding the polygonal interpolation of the points. Polygonal representations have two major difficulties, however. Vertices concentrate unstably around high curvature regions, which makes matching more difficult. More importantly, the only information that is available is the length of the segments and the angles between the segments. We are more interested in differential invariants along the curves, and thus favor higher-order spline approximations.

B-spline curves, which includes the class of polygonal curves, offer good approximation properties (see [JAJ67]), can readily provide differential information at any point along the spline curve, and satisfy certain optimality properties, viz., they minimise a certain measure of the bending energy [Hol57]. There is an extensive literature on B-splines [JAJ67, BBB87, dB78, Far88, Cin81]. A survey is provided in [BFK84]. We provide a very brief introduction, using the notation of [BBB87, SMM90].

Given a sequence of $n+1$ points $\mathbf{P_j}(x_i, y_i, z_i)$, $i = 0..n$ in 3-space, a C^{K-2} approximating B-spline consists of the following components:

1. A control polygon of $m+1$ points is given, such that $\mathbf{V_j}(\mathbf{X_j}, \mathbf{Y_j}, \mathbf{Z_j}), j = 0..m$ are known points;

2. We are given $m+1$ real-valued piecewise polynomial functions, $B_{j,K}(\bar{u})$, representing the basis splines, which are functions of the real variable $\bar{u}$ and consist of polynomials of degree $K-1$, and are globally of class C^{K-2} (so that $K-2$ derivatives match at the knots). The location in 3-space of the approximating curve for a given parameter value $\bar{u}$ is given by:

$$\begin{aligned}\mathbf{Q}(\bar{u}) &= (X_q(\bar{u}), Y_q(\bar{u}), Z_q(\bar{u})) = \sum_{j=0}^{m} \mathbf{V_j} B_{j,K}(\bar{u}) \\ &= \sum_{j=0}^{m} (X_j B_{j,K}(\bar{u}), Y_j B_{j,K+1}(\bar{u}), Z_j B_{j,K}(\bar{u});\end{aligned}$$

3. The knots must also be specified, and consist of $m+K$ real values $\{\bar{u}_j\}$, with $\bar{u}_1 = 0$ and $\bar{u}_{m+K} = L$, partitioning the interval $[0, L]$ into $m+K-1$ intervals. Here, L is the length of the polygon joining the $\mathbf{P_i}$'s. If the intervals are uniform, then we say that the approximation is a uniform B-spline.

We use the global parameter $\bar{u}$ along the interval $[0, L]$, and denote by u the relative distances between knots, defined by $u = (\bar{u} - \bar{u}_i)/(\bar{u}_{i+1} - \bar{u}_i)$. The basis spline functions are defined recursively. The basis splines of degree 1 are simply the characteristic functions of

the intervals:

$$B_{j,1}(\bar{u}) = \begin{cases} 1 & \bar{u}_j \leq \bar{u} < \bar{u}_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

Successively higher-order splines are formed by blending lower-order splines:

$$B_{j,K+1}(\bar{u}) = \frac{\bar{u} - \bar{u}_i}{\bar{u}_{i+K} - \bar{u}_i} B_{j,K}(\bar{u}) + \frac{\bar{u}_{i+K+1} - \bar{u}}{\bar{u}_{i+K+1} - \bar{u}_{i+1}} B_{j+1,K}(\bar{u}).$$

It is not hard to show that:

$$\frac{\partial B_{j,K+1}(\bar{u})}{\partial \bar{u}} = K[\frac{B_{j,K}(\bar{u})}{\bar{u}_{j+K} - \bar{u}_j} - \frac{B_{j+1,K}(\bar{u})}{\bar{u}_{j+K+1} - \bar{u}_{j+1}}].$$

Thus quadratic splines, the $B_{j,3}$, are C^1, cubic splines ($B_{j,4}$), are C^2, etc. Derivatives of the approximating B-spline can be written in an especially simple fashion, involving vector directions that depend on scaled differences of the control points $\mathbf{V}_j$:

$$\begin{aligned} \frac{\partial^r \mathbf{Q}}{\partial \bar{u}^r} &= \sum_{j=0}^{m} \mathbf{V}_j B_{j,K}^{(r)}(\bar{u}) \,, \\ &= \sum_{j=r}^{m} \mathbf{V}_j^r B_{j,K+1-r}(\bar{u}) \,, \; \mathbf{V}_j^0 = \mathbf{V}_j \,, \\ \mathbf{V}_j^r &= (K-r) \frac{\mathbf{V}_j^{r-1} - \mathbf{V}_{j-1}^{r-1}}{\bar{u}_{j+K-r+1} - \bar{u}_j}. \end{aligned}$$

Because of this simple formula, we may incorporate contraints on the derivatives in our measure of the quality of an approximation, for the process of finding the best control points and knots, and we will also be able to easily make use of differential measures of the curve for matching purposes.

Given any smooth space curve $\mathbf{Q}(\bar{u})$, the curvature and torsion of the curve at a point $\bar{u}$ are defined by:

$$k(\bar{u}) = \frac{||\mathbf{Q}'(\bar{u}) \wedge \mathbf{Q}''(\bar{u})||}{||\mathbf{Q}'(\bar{u})||^3}, \; \tau(\bar{u}) = \frac{\det(\mathbf{Q}'(\bar{u}), \mathbf{Q}''(\bar{u}), \mathbf{Q}'''(\bar{u}))}{||\mathbf{Q}'(\bar{u}) \wedge \mathbf{Q}''(\bar{u})||^2}$$

Here, $\wedge$ denotes the vector cross product. The Frénet frame $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ is defined as follows:

$$\mathbf{t}(\bar{u}) = \frac{\mathbf{Q}'(\bar{u})}{||\mathbf{Q}'(\bar{u})||}, \; \mathbf{b}(\bar{u}) = \frac{\mathbf{Q}'(\bar{u}) \wedge \mathbf{Q}''(\bar{u})}{k(\bar{u})||\mathbf{Q}'(\bar{u})||^3},$$

and $\mathbf{n}$ is defined as the vector cross product of $\mathbf{b}$ and $\mathbf{t}$.

2.2 A previous approximation scheme

We next recall a classic approximation scheme due to Barsky [BBB87]. This scheme has been used by St-Marc and Médioni [SMM90] for curve matching. Our emphasis is on the shortcomings of the approach for our objectives and on proposed modifications.

It is known that uniform B-splines are optimal in a certain least squares sense. This can be justified by the supposition that the positions of the data points lie on a polynomial

curve, perturbed by identically distributed Gaussian noise. A least squares derivation then corresponds to a maximum likelihood reconstruction of the data [LS86].

Given $n+1$ data points $\mathbf{P_i}(x_i, y_i, z_i), i = 0..n$, we seek $m+1$ control vertices $\mathbf{V_j}$, $j = 0..m$ and $m + K$ corresponding knots $\bar{u}_j$, $j = 0..m + K$ minimizing the sum of square distances between the B-spline $\mathbf{Q}(\bar{u})$ of degree $K-1$ and the data $\mathbf{P_i}$. The notion of distance between a spline $\mathbf{Q}(\bar{u})$ and a data point $\mathbf{P_i}$ is based on the parameter value $\tilde{u}_i$ where the curve $\mathbf{Q}(\bar{u})$ comes closes to $\mathbf{P_i}$. Thus, the criterion is to minimize:

$$\begin{aligned}
\Delta_1 &= \sum_{i=0}^{n} ||\mathbf{Q}(\tilde{u}_i) - \mathbf{P_i}||^2 \\
&= \sum_{i=0}^{n} [(X_q(\tilde{u}_i) - x_i)^2 + (Y_q(\tilde{u}_i) - y_i)^2 + (Z_q(\tilde{u}_i) - z_i)^2] \\
&= \sum_{i=0}^{n} [(\sum_{j=0}^{m} X_j B_{j,K}(\tilde{u}_i) - x_i)^2 + (\sum_{j=0}^{m} Y_j B_{j,K}(\tilde{u}_i) - y_i)^2 \\
&+ (\sum_{j=0}^{m} Z_j B_{j,K}(\tilde{u}_i) - z_i)^2]
\end{aligned}$$

The calculation of the $\tilde{u}_i$ values is critical, since $||\mathbf{Q}(\tilde{u}_i) - \mathbf{P_i}||$ is supposed to represent the Euclidian distance of the point $\mathbf{P_i}$ to the curve. On the other hand, an exact calculation of the values $\tilde{u}_i$ is difficult, since they depend implicitly on the solution curve $\mathbf{Q}(\bar{u})$. As an expedient, Barsky suggests using for $\tilde{u}_i$ the current total length of the polygonal curve from $\mathbf{P_0}$ to $\mathbf{P_i}$, recalling that $0 \leq \bar{u} \leq L$, and L is the length of the entire polygonal path. Thus as an estimate, we can use $\tilde{u}_i = \sum_{k=0}^{i-1} ||\mathbf{P}_{k+1} - \mathbf{P}_k||$.

Since the sum of errors Δ_1 to be minimized is quadratic, it is easy to show that the collection of unknowns $\mathbf{V_j}(X_j, Y_j, Z_j)$, $j = 0..m$ satisfy:

$$\begin{aligned}
\frac{\partial \Delta_1}{\partial X_j} &= 2 \sum_{i=0}^{n} B_{j,K}(\tilde{u}_i)(\sum_{l=0}^{m} X_l B_{l,K}(\tilde{u}_i) - x_i) = 0, \\
\frac{\partial \Delta_1}{\partial Y_j} &= 2 \sum_{i=0}^{n} B_{j,K}(\tilde{u}_i)(\sum_{l=0}^{m} Y_l B_{l,K}(\tilde{u}_i) - y_i) = 0, \\
\frac{\partial \Delta_1}{\partial Z_j} &= 2 \sum_{i=0}^{n} B_{j,K}(\tilde{u}_i)(\sum_{l=0}^{m} Z_l B_{l,K}(\tilde{u}_i) - z_i) = 0.
\end{aligned}$$

As j ranges from 0 to m, we obtain 3 linear systems of size $m+1$ for the unknowns X_j,Y_j and Z_j. For the X_j variables, for example, we have the system:

$$\begin{cases}
\sum_{j=0}^{m} [\sum_{i=0}^{n} B_{0,K}(\tilde{u}_i) B_{j,K}(\tilde{u}_i)] X_j &= \sum_{i=0}^{n} x_i B_{0,K}(\tilde{u}_i) \\
\vdots & \vdots \\
\sum_{j=0}^{m} [\sum_{i=m}^{n} B_{m,K}(\tilde{u}_i) B_{j,K}(\tilde{u}_i)] X_j &= \sum_{i=m}^{n} x_i B_{m,K}(\tilde{u}_i)
\end{cases}$$

If the approximating curve is not a closed curve, then we require that the first and last data control points coincide with the first and last data points, thereby suppressing two degrees of freedom for each variable. For a closed curve, the final $K-1$ control vertices are also the first $K-1$ vertices, with corresponding knot discrepancies.

If we write the three linear systems in the form $AX = B_x$, $AY = B_y$, $AZ = B_z$, we discover that A is a symmetric banded matrix with $2K-3$ nonzero bands (for an nonclosed curve). For a closed curve approximation, extra nonzero values appear in the upper right and lower left corners of the matrix.

As an example, consider the case where we desire seven control vertices. The system then has form:

$$\begin{pmatrix} \sum_{i=0}^{n} B_{1,K}(\tilde{u}_i)^2 & B_{1,K}B_{2,K} & B_{1,K}B_{3,K} & 0 & 0 \\ \sum_{i=0}^{n} B_{2,K}(\tilde{u}_i)B_{1,K}(\tilde{u}_i) & B_{2,K}^2 & B_{2,K}B_{3,K} & B_{2,K}B_{4,K} & 0 \\ B_{3,K}B_{1,K} & B_{3,K}B_{2,K} & B_{3,K}^2 & B_{3,K}B_{4,K} & B_{3,K}B_{5,K} \\ 0 & B_{4,K}B_{2,K} & B_{4,K}B_{3,K} & B_{4,K}^2 & B_{4,K}B_{5,K} \\ 0 & 0 & B_{5,K}B_{3,K} & B_{5,K}B_{4,K} & B_{5,K}^2 \end{pmatrix}$$

Thus, given the approximate locations of the $\tilde{u}_i$ values as suggested by Barsky, the matrix system A may be constructed in linear time (in the number of data points), and the systems may be solved, for example, by a Crout factorization or, more efficiently, using a Choleski decomposition. Numerical instabilities can appear in the case when a model has a small number of control vertices (e.g. consists of a straight line).

In working with this method, we have observed that m, the number of control points, must be quite large in order to obtain a good *visual* fit to the data points. Worse, small amplitude oscillations often appear, corrupting the derivative information, and making derivative-based matching methods unworkable. For example, using the synthetic data of a noisy helix (Fig. 3), we reconstruct Fig. 4 using the Barsky method for spline approximation. It can be seen that curvature and torsion measurements along the approximation curve will be unstable. In the next section, we explain how the results shown in Figs. 5 and 6 are obtained.

2.3 Improvements

2.3.1 Better knot distribution

The vertices of an approximating polygonal path will concentrate around locations of high curvature [Pav77]. We make use of this property to distribute B-Spline knots non-uniformly with respect to segment lengths, so that the knots are denser around high curvature points. In this way, the B-spline will more closely approximate these portions of the curve. In order to obtain the initial polygonal approximation, we use the classical recursive algorithm of Duda and Hart [DH73] which can be easily stated: Initially, a single line joins the two endpoints of the curve; then iteratively, a vertex is added at the point of the curve of maximum distance from the polygonal path.

The number of knots m is also determined by the number of vertices. However, we utilize the following approach to locate the initial placements of the points representing the locations of closest approach to the data points, $\tilde{u}_i$: Rather than following Barsky's suggestion (which makes use of the interpolating polygonal path, as opposed to the approximating polygonal path), we simply project each point $\mathbf{P_i}$ onto the approximating polygonal path and considering the relative position of the projected points in terms of total chordlength of the path (see B in Fig. 7).

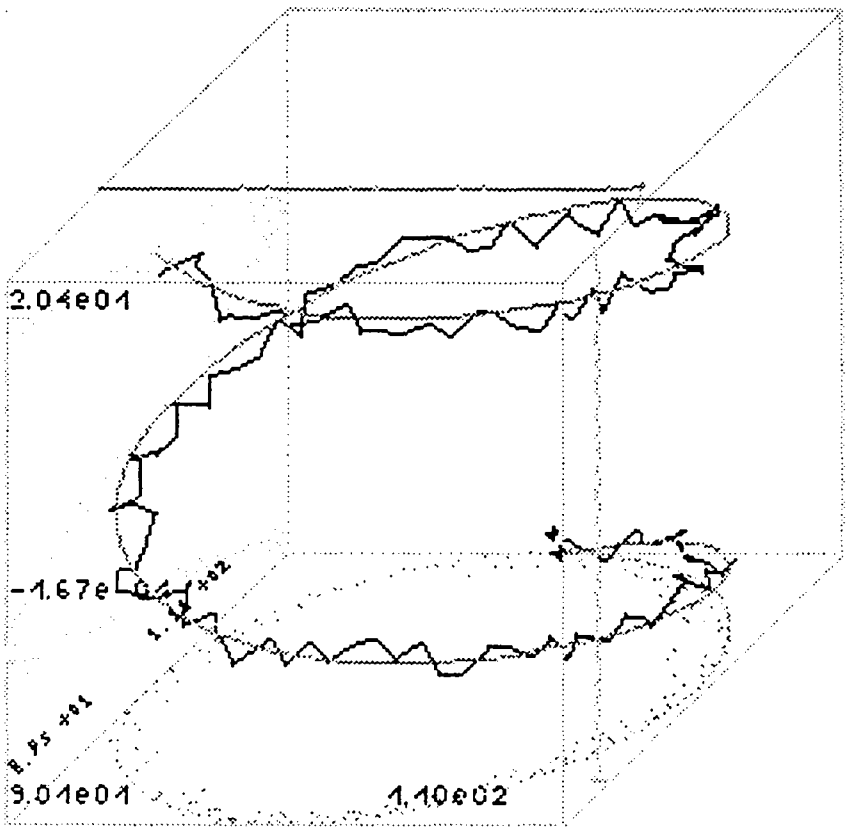

Figure 3: Noise is added to a helix, and points are sampled with a limitation on the distance between successive points. The curvature and torsion are plotted in the top and the right panels of the cube, as a function of arclength. In a perfect reconstruction, the curvature and torsion would be constant.

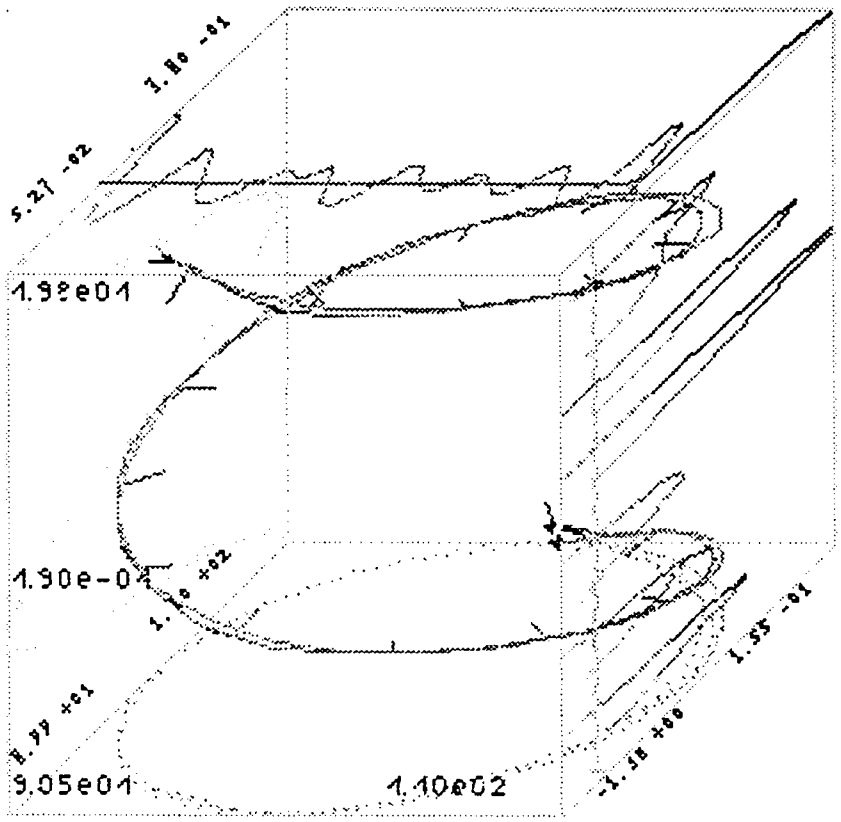

Figure 4: In the reconstruction method as suggested by Barsky, curvature and (especially) torsion values are extremely noisy, despite the quality of the reconstruction (in terms of position) of the original curve.

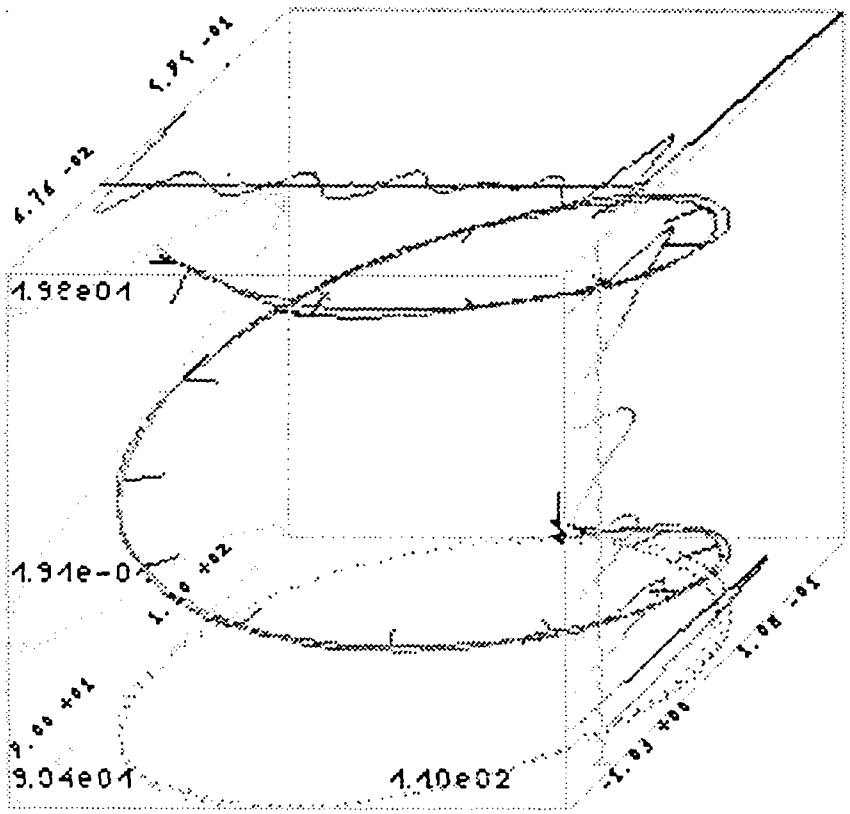

Figure 5: A more precise estimate of model-data distances improves the estimation of curvature and torsion.

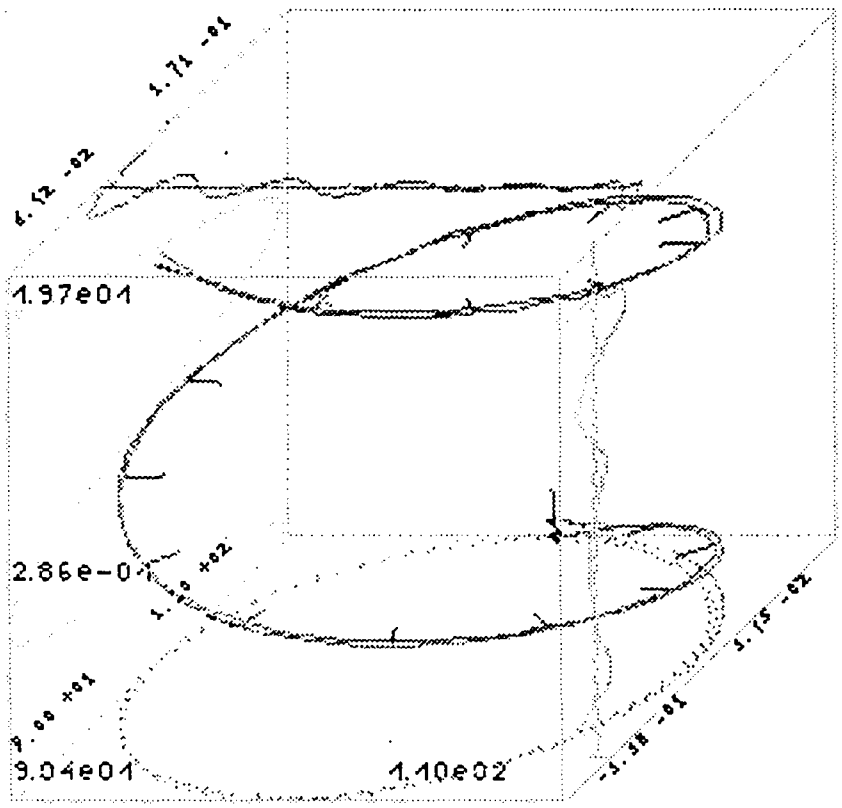

Figure 6: The constraint on the second derivative also improves the estimation.

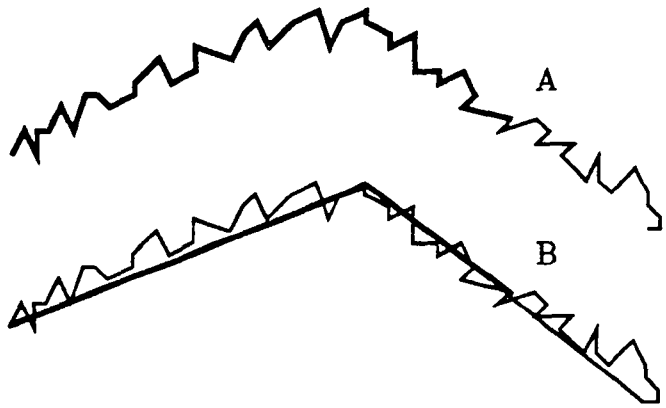

Figure 7: A comparison of the initial positions for the projected point locations $\tilde{u}_i$, represented as distances (thick line) on a polygon using Barsky's method (A) and our modification (B).

This method, projection onto the approximating polygonal path, seems to outperform Barsky's proposal (see Fig. 7), especially when data points are densely populated around the curves.

2.3.2 Improved distance estimates [PS83]

Estimation of the $\tilde{u}_i$ values in Fig. 7 is based on the following hypotheses:

1. The global parameter $\bar{u}$ behaves as arclength, and thus the curve is traversed with constant speed. However, the derivatives of the spline will depend on control vertex positions. Thus, all other parameters being equal, if distance increases between successive control points, so does the speed of the parameterization.

2. The arclength is well approximated by the length along the polygon joining the data. This hypothesis is valid in the sense that the arclength converges to the true value as the number of vertices increases, but discretization errors and especially noise will cause errors, whereas our aim is to accurately smooth the noise and achieve a continuous representation.

Accordingly, the projection method has advantages, but can still lead to problems.

Thus we next study the distance between a point and a polynomial curve of arbitrary degree. The true $\tilde{u}_i$ corresponds to the minimum of $||\mathbf{Q}(\tilde{u}_i) - \mathbf{P_i}||$. Let us thus consider the following equation, where $\tilde{u}_i$ is unknown:

$$F_i(\tilde{u}_i) = \frac{\partial ||\mathbf{Q}(\tilde{u}_i) - \mathbf{P_i}||}{\partial \tilde{u}_i} = 0 = \frac{\partial ||\sum_{j=0}^{m} V_j B_{j,K}(\tilde{u}_i) - \mathbf{P_i}||}{\partial \tilde{u}_i}$$
$$= 2(\sum_{j=1}^{m} \frac{(K-1)(V_j - V_{j-1})}{\bar{U}_{j+K-1} - \bar{U}_j} B_{j,K-1}(\tilde{u}_i)) \cdot (\sum_{j=0}^{m} V_j B_{j,K}(\tilde{u}_i) - \mathbf{P_i}).$$

We update $\tilde{u}_i$ by Newton Raphson iteration, using the quantity $\delta_i = F_i(\tilde{u}_i)/F_i'(\tilde{u}_i)$. Defining

$$DV_j = \frac{(K-1)(V_j - V_{j-1})}{\bar{U}_{j+K-1} - \bar{U}_j}; \quad D^2V_j = \frac{(K-2)(DV_j - DV_{j-1})}{\bar{U}_{j+K-2} - \bar{U}_j},$$

(with similar equations for each component X_j, Y_j, and Z_j), we obtain

$$\begin{aligned}
F_i(\tilde{u}_i) &= (\sum_{j=1}^{m} DX_j B_{j,K-1}(\tilde{u}_i)) \cdot (\sum_{j=0}^{m} X_j B_{j,K}(\tilde{u}_i) - x_i) \\
&+ (\sum_{j=1}^{m} DY_j B_{j,K-1}(\tilde{u}_i)) \cdot (\sum_{j=0}^{m} Y_j B_{j,K}(\tilde{u}_i) - y_i) \\
&+ (\sum_{j=1}^{m} DZ_j B_{j,K-1}(\tilde{u}_i)) \cdot (\sum_{j=0}^{m} Z_j B_{j,K}(\tilde{u}_i) - z_i), \\
F_i'(\tilde{u}_i) &= (\sum_{j=2}^{m} D^2 X_j B_{j,K-2}(\tilde{u}_i)) \cdot (\sum_{j=0}^{m} X_j B_{j,K}(\tilde{u}_i) - x_i) \\
&+ (\sum_{j=2}^{m} D^2 Y_j B_{j,K-2}(\tilde{u}_i)) \cdot (\sum_{j=0}^{m} Y_j B_{j,K}(\tilde{u}_i) - y_i) \\
&+ (\sum_{j=2}^{m} D^2 Z_j B_{j,K-2}(\tilde{u}_i)) \cdot (\sum_{j=0}^{m} Z_j B_{j,K}(\tilde{u}_i) - z_i) \\
&+ (\sum_{j=1}^{m} DX_j B_{j,K-1}(\tilde{u}_i))^2 + (\sum_{j=1}^{m} DY_j B_{j,K-1}(\tilde{u}_i))^2 \\
&+ (\sum_{j=1}^{m} DZ_j B_{j,K-1}(\tilde{u}_i))^2.
\end{aligned}$$

Despite the apparent complexity of these equations, these computations are not very expensive, since $B_{j,K-1}$ et $B_{j,K-2}$ were necessarily calculated before $B_{j,K}$ (by the recursive definition). Moreover, once all $\tilde{u}_i$ are updated by the amounts δ_i, (the Newton-Raphson correction amount), we must once again solve the linear system for new control vertices $\{V_j\}$. Fig. 8 illustrates the idea. We may repeat this operation until the solution satisfies the ϵ_∞ criterion or until no substantial decrease of the maximum error is observed.

Fig. 5 shows the improvement that iterations effect on the appoximation curve.

Up to now in this section, we have only been concerned with the placement of the knots and the parameter values $\tilde{u}_i$ associated with each point that is used to define the spline approximation. In the two next subsections, we address the problem of refining the approximation and the control point positions so as to obtain good model regularization by constraining the first and second derivatives of the curve.

2.3.3 Minimisation of curvature

Cubic B-splines have an optimal approximation property [JAJ67], namely that among all interpolants they minimise the norm of the second derivative. This result was first proven by Holladay [Hol57]. Alternative criteria can be posed for smoothing; for example, we might choose to minimize a weighted sum of the squared second derivatives of the approximating curve (evaluated at the projection points) together with the distance error from the data points:

$$\Delta_2 = \frac{\Delta_1}{{\sigma_1}^2} + \frac{\sum_{i=0}^{i=n} ||\mathbf{Q}''(\tilde{u}_i)||^2}{{\sigma_2}^2},$$

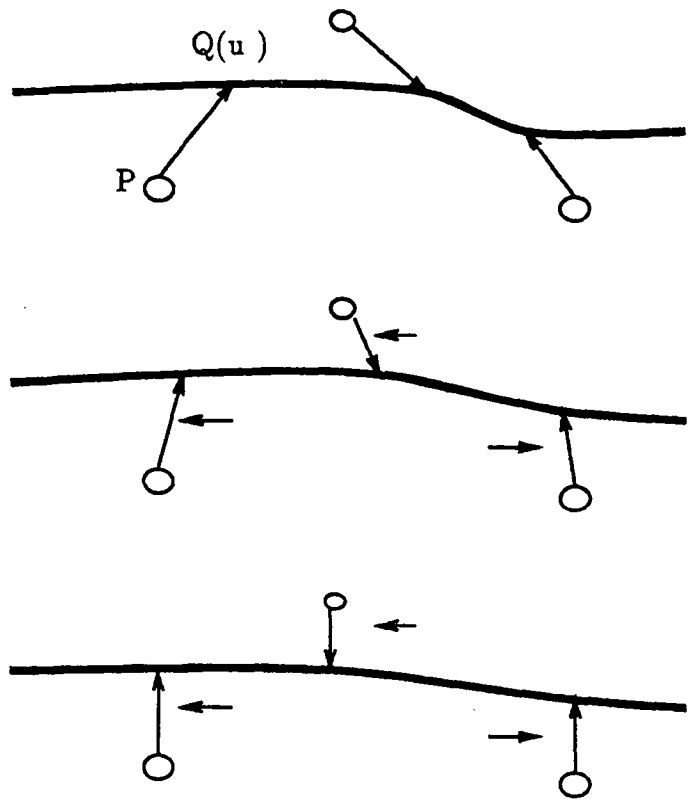

Figure 8: For each point P the corresponding $\tilde{u}$ follows a gradient descent. The least squares solution is modified at each step.

where

$$\begin{cases} {\sigma_1}^2 &= \text{Var}(||\mathbf{Q}(\tilde{u}_i) - \mathbf{P_i}||), \\ {\sigma_2}^2 &= \text{Var}(||\mathbf{Q}''(\tilde{u}_i)||), \end{cases}$$

and Var designates the observed variance of the argument values over the index i.

The second term is related to the bending energy of the spline. Since the second derivative values are linear in terms of control vertices, Δ_2 is again quadratic. We solve a linear system of size $m+1$, and the construction and complexity are as before, and the result is a spline. In fact, the complexity will be linear in terms of the number of data points and cubic in terms of the number of control vertices. For comparison [Arb90] constrains the $(K-1)$st derivative with a tension parameter. Fig. 6 illustrates results of minimizing Δ_2.

2.3.4 Incorporation of surface normals

Finally, we assume that the curve is supposed to lie in a surface whose normals are known. Thus, at every point along the approximating curve, the tangent direction should lie normal to the surface normal $\mathbf{n_i}$. Accordingly, we penalize our optimization criterion by a measure of the violations of this condition:

$$\Delta_3 = \Delta_2 + \frac{\sum_{i=0}^{i=n} (\mathbf{Q}'(\tilde{u}_i).\mathbf{n_i})^2}{{\sigma_3}^2},$$

with :

$$\sigma_3{}^2 = \mathrm{Var}(\mathbf{Q}'(\tilde{u}_i) \cdot \mathbf{n}_i).$$

Note that the surface normals are a function of position, and must be provided in all of three-space (or in any case, near the surface), even though the normal vector field is only truly defined on the surface. This is the case when dealing with 3D medical images including (possibly noisy) iso-intensity surfaces. The gradient of the intensity function is identified with the surface normal direction, and is available at any 3D point. Also, note that although the surface normal is NOT in general identical to the curve normal, it is always orthogonal to the curve tangent, establishing the validity of our Δ_3 criterion. Finally, Δ_3 is still quadratic, but due to the scalar product, variables cannot be separated and the system size is multiplied by three.

The progress obtained by minimizing Δ_3 can be shown using real scanner data, with the aim of finding the rigid transformation between two views of the same object (a skull) from characteristic curves (ridges). In Fig. 9 we show two such ridge curves on one view of a skull, specifically the chin curve and orbital ridge curve, which will permit a matching accuracy of one or two voxels (our representation of the chin curve is presented in Fig. 10). The two ridges are also present in a second view of the skull, with an occlusion of part of the chin curve, and a small shift of the (perceptual) origin of the chin. There are no obvious matching pairs of points between the two views (Fig. 11). Using surface normals in Fig. 12, the curvature values of the chin curve in the second view are closer to those of the first view. The torsion function in the second view has an abrupt negative spike, which is one reason why we investigate more stable invariants using lower-order derivatives in Section 3.

In summary, smoothing with regularized B-splines can be done in linear time in the number of data points and cubic time in the number of control vertices, and so is $O(n+m^3)$. It leads to a minimum of energy with a direct solution possessing stable differential parameters up to the second order, in real time on a standard workstation.

3 Recognition and localization

Having represented a curve by a B-spline approximation, we now wish to match an extracted curve with a model curve. By *localization*, we mean the accurate positioning of the curve and its features relative to a matching model curve. We will make use of the analytic structure of the approximating spline. Formally, our problem is stated as follows: We are given a set of model curves $\{M_i\}$ and an extracted (unknown) 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; and
- To specify that rigid transformation that best associates the two curves.

There are numerous approaches to this problem. In order to best compare our method with other approaches, we begin by recalling a couple of classes of alternatives.

All approaches make use of features extracted along the curve, and make fundamental use of the fact that an intrinsic reference frame can be associated with each point along the

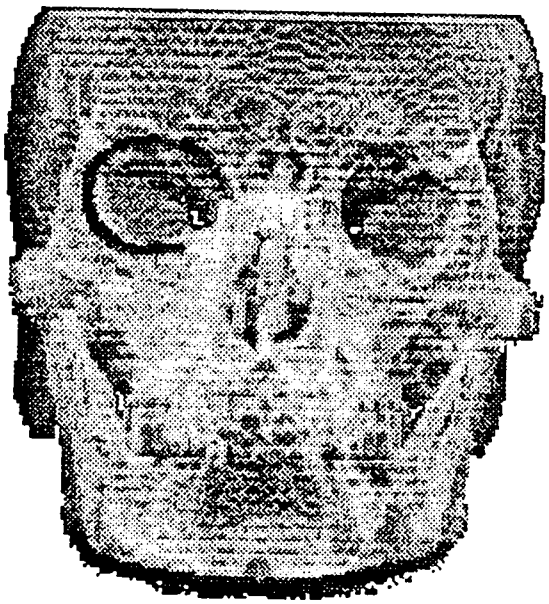

Figure 9: Ridges on skull (first view): the chin curve and orbital ridge.

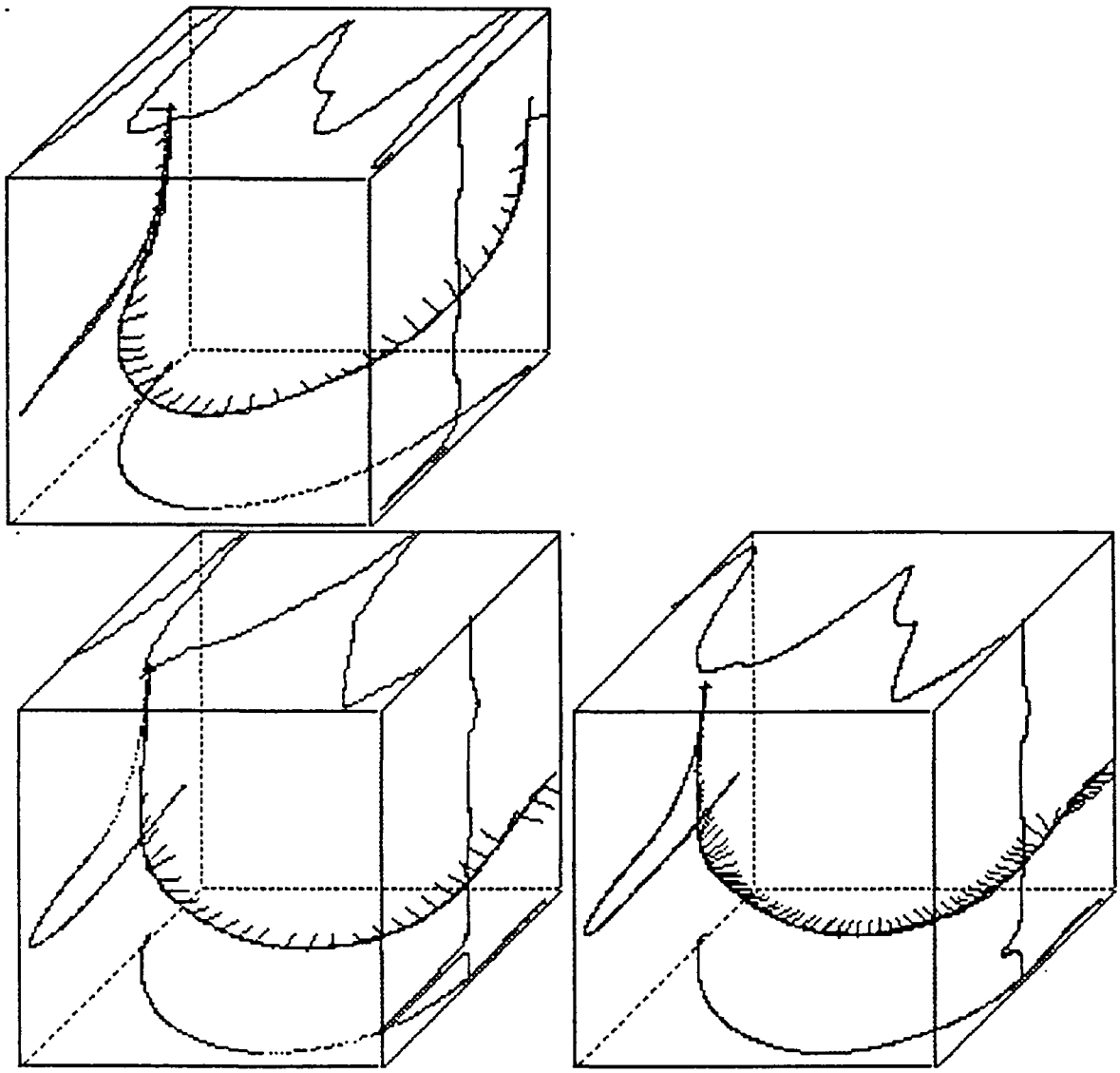

Figure 10: Approximation of the chin curve by our method. We plot curvature (top),and torsion (right) as a function of arclength, as well as Frénet frames along the curve.

Figure 11: The chin curve extracted from a second view of the skull; note that part of the curve is occluded.

Figure 12: With the help of surface normals, the curvature representation is much more reliable (The centered double peak is in both views).

model curve (the Frénet frame). Accordingly, given a correspondence between two points on two respective curves, a unique rigid transformation $D = (R, \mathbf{u})$ can be defined that maps one point to the other in such a way that the corresponding coordinate frames are aligned.

Specifically, given a pair of points (M, S), point M belonging to a model and point S belonging to an extracted curve, and also given the associated Frénet frames $(\mathbf{t}, \mathbf{n}, \mathbf{b})$ and $(\mathbf{t}', \mathbf{n}', \mathbf{b}')$, the rotation R which brings the two frames into correspondence is given simply by the outer product:

$$R = (\mathbf{t}, \mathbf{n}, \mathbf{b})^t (\mathbf{t}', \mathbf{n}', \mathbf{b}').$$

For the translational component, suppose that O is the origin in a global reference frame, and that $\mathbf{OS}$ denotes the vector from O to S, while $\mathbf{OM}$ is the vector from O to M, then

$$\mathbf{u} = \mathbf{OS} - R(\mathbf{OM}).$$

We now consider, in general terms, the complexity of three methods of model matching. A more detailed complexity analysis would require a closer study of the costs of sampling the data along the curves, which we do not pursue here. However, under the assumption that the number of points on each curve is of the order of n, the following analysis suffices to indicate relative costs. In all cases, we assume that every curve is represented by a sequence of samples, parameterized by arclength s, with a Frénet frame, and thus curvature $c(s)$ and torsion $\tau(s)$ are available at each sample point along each curve.

3.1 Approaches to curve model matching

3.1.1 Prediction-Verification

A subset of k significant points of the curve S are selected according to some stable criterion. For example, points of curvature extrema might be used. Choosing one of the k points, we then attempt to match that point with a model M_i and a point along M_i having a similar curvature and torsion. Each such matching constitutes a hypothesis.

Since a Frénet frame is associated with both of the points used in a hypothesis, a rigid transformation can be calculated. In this way, the hypothesis can be checked. Initially, only neighboring points in the vicinity of the matched points are checked, to see whether the transformation brings them into proximity. If so, a revised transformation may be computed using a least-mean-squares approach, iteratively improving the hypothesis, for example by making use of a Kalman filter [Aya91].

By checking or all possible hypotheses, we find the pair which leads to the greatest number of correspondences.

The algorithmic complexity of this approach is proportional to the number of models. For each model, we must iterate over all k selected points in the scene, and the complexity per model lies somewhere between $O(kn \log n)$ and $O(kn^2 \log n)$. The is because for every selected point on the curve S, roughly $\log n$ comparisons are needed in order to select the candidate points in a single model with similar curvature and torsion values (assuming the data in the model has been sorted in a preprocessing phase). Then, the number of verifications that will be required is anyone between 1 and n, depending on the degree of discrimination effected by the curvature and torsion, and each verification step is $O(n)$.

3.1.2 Accumulation of parametric evidence (Hough techniques)

For each and every point on the curve S and every point on a model M_i having a similar curvature and torsion, we suppose that the two points are matched. For each such association, we compute the rigid transformation as defined by the corresponding Frénet frames. The rigid transformation determines six parameters, and we register a vote in a quantized six-dimensional parameter space. If the curve and model match, then there should be a bin in the parameter space receiving a lot of votes, corresponding to the rigid transformation bringing the two curves into correspondence. The process is repeated for every model.

The complexity of this algorithm is proportional to the number of models. For each model, the complexity is between $O(n \log n)$ et $O(n^2 \log n)$, depending on the degree of discrimination introduced by the curvature and torsion values.

3.1.3 Model indexing

In a preprocessing phase, we construct an *indexation table*, where entries are associated with pairs of values (c, τ). For each pair, a list of entries of the form $m_{i,j}$ is formed, denoting the fact that point number j on model M_i has a curvature and torsion value that is close to (c, τ). Note that we assume that the models curves have been sampled.

During the recognition phase, we walk along the list of points of S, and for each point s_l we examine the list of entries associated with the index $c(s_l)$, $\tau(s_l)$. For each entry $m_{i,j}$ in the list, we compute a six-parameter rigid transformation $D_{i,j,l}$ that would bring the point on S at s_l into correspondence with the point $m_{i,j}$ of model M_i. We register a vote for the pair $(M_i, D_{i,j,l})$. Note that the first parameter is discrete, whereas the second lies in a six-dimensional continuous space.

After processing all of the points along S, we locate the pairs of the form (model, displacement) that have received a lot of votes (relative to some error measure in displacements), and verify the indicated matches.

The complexity of the recognition phase, disregarding the preprocessing phase, is essentially independent of the number of models. The apparent complexity lies somewhere between $O(n)$ and $O(n^2)$, depending on the level of quantization of the index space according to curvature and torsion. We assume that there are at least as many bins as models, and that the entries are sufficiently distributed such that the length of a bin's entry is at the worst $O(n)$. The method depends on the existence of variation along the model curves; for example, the method will not work for recognition of a helix, since the curvature and torsion is constant along a helix.

This description of the method of indexation is essentially the "geometric hashing" method of Kishon and Wolfson [KHW89], updated in one important respect. They use a polygonal representation of the curves, and thus vote for a model and a displacement length, representing a difference between the arclength 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 include the explicit calculation of the entire rigid transformation as part of the recognition process. The advantage of our method is that the arclength parametrization can suffer from inaccuracies and accumulative errors, whereas the

six-parameter rigid transformation suffers only from local representation error. For example, consider a model curve which suffers from a number of insertion errors (which might arise from frequent local perturbations along the curve, due, say, to contact between a surface and some other object, or due to a clamp or other surface artifact). In using the arclength parameterization, each matching subsection will have to be recognized separately, voting for the same model but with different arclength displacements. With the differentiable representation, each subsection may vote for the correct model with the a uniform (approximate) rigid transformation. (Our approach, while different, may also be compared with that of Stein [Ste91].)

Another advantage of voting for rigid transformations is that we may use a statistical method to compute a distance between two such transformations, and incorporate this into the voting process and the indexation table. We explain this enhancement in greater detail below, and also show how the various transformations $D_{i,j,l}$ voting for a winning candidate model can be combined for the purposes of verifying the match.

3.2 Enhancements to the indexation method

Since the indexation method offers computational advantages, especially in terms of its sublinear complexity growth in the number of models, we investigate enhancements and implementation issues for an application in medical image processing.

The following subsections detail the modifications to the basic indexation method that we have employed.

3.2.1 Indexation table quantization

We collect statistics based on experiments with simulation and real data, described in more detail in Section 3.2.4. These statistics provide expected variances for the curvature and torsion values of typical noisy curves, and also covariance values for pairs of values taken from intra- and inter- curve pairs of points. In order to establish an "optimal" discretization cell size in the (c, τ) space, we study these covariance values. The cell size must be sufficiently large in order to account for the expected noise that arises due to normal variations, but sufficiently small in order to provide discrimination between different points along a typical curve.

3.2.2 A metric for rigid transformations

At the same time, we compute covariance values for the six-parameter rigid transformations that are obtained by matching points along a scene curve with model curves. The resulting covariance matrix is used in the definition of the Mahalanobis distance metric which we subsequently use to determine the proximity of two distinct rigid transformations, used during the recognition phase of the indexation algorithm.

3.2.3 Recursive transformation estimation

Throughout the recognition phase, as soon as a pair of points are matched such that the transformation defined by the associated Frénet frames is sufficiently close to some previously recognized matching, the estimation of the prototype transformation to be used as the matching criterion may be refined through the use of a recursive filter, such as the Kalman filter. The experiments show that this procedure can significantly improve the robustness of the method.

3.2.4 Alternative geometric invariants for matching

We were led by the statistical study of noisy curves to search for invariant parameters that can serve as alternatives to the curvature and torsion measures that are used for indexing. Suppose that we are given a reference point B on a model curve, and consider the points P on the same curve (see Fig. 13). For each point P, we can define the rigid transformation $D = (R, \mathbf{u})$ that maps the Frénet frame at B onto the Frénet frame at P, and associate the six parameters with the point P. For a fixed basis point, these parameters are invariant with respect to rigid transformations, and consist of the three rotation coordinates $(\mathbf{r}_t, \mathbf{r}_n, \mathbf{r}_b)$ with respect to the basis frame, [1] and the translation coordinates $(\mathbf{u}_t, \mathbf{u}_n, \mathbf{u}_b)$, again measured in the basis frame. If the curve lies in a plane (a planar curve), then $\mathbf{r}_t$ will always be zero, in which case it is preferable to use the representation $(\theta_t, \theta_n, \theta_b)$, angles between the vectors of the frame at B and of the frame at P.

To investigate the utility of these various invariants, we took a single curve of discrete data (from real data), and formed a database of 100 random transformations of this curve. For each transformation, we applied the spline smoothing and representation algorithm of Section 2, and then sample each spline curve with a uniform sampling. A single corresponding basis point is identified on all of the curves, and other homologous points P may be determined from the sample number. For each such point P, we compute the curvature, torsion, and other invariants, as discussed above. The inter-curve variances are computed from the set of all pairs of corresponding points P on different curves, and the intra-curve variances are computed using distinct points P along a single curve. Variances are computed for every invariant. The quotient between the inter-curve variance and the intra-curve variance measures the stability of the corresponding invariant. The results of this study are summarized in the following table:

σ	c	τ	$\|\|\mathbf{u}\|\|$	θ_t	θ_n	θ_b	$\mathbf{u}_t$	$\mathbf{u}_n$	$\mathbf{u}_b$
inter	0.0034	0.4069	0.8932	0.0368	0.0832	0.0978	0.8267	1.0089	2.4657
intra	0.0192	0.5137	31.774	0.8822	0.5479	0.6602	36.350	8.8191	38.836
ratio	**0.17771**	*0.7921*	0.0281	**0.0417**	0.1518	0.1477	**0.0227**	0.1143	0.0634

We observe that the invariants θ_t and $\mathbf{u}_t$ are more stable than torsion, and have greater discrimination power than $||\mathbf{u}||$. We expect to make use of the parameters $(c, \theta_t, \mathbf{u}_t)$ as the index values in a new indexation algorithm that we describe in the next subsection.

[1] Recall that a rotation in R^3 can be parameterized by a vector r whose direction corresponds to the axis of rotation, and whose length represents the angle of rotation about that axis. (Consult [Aya91].)

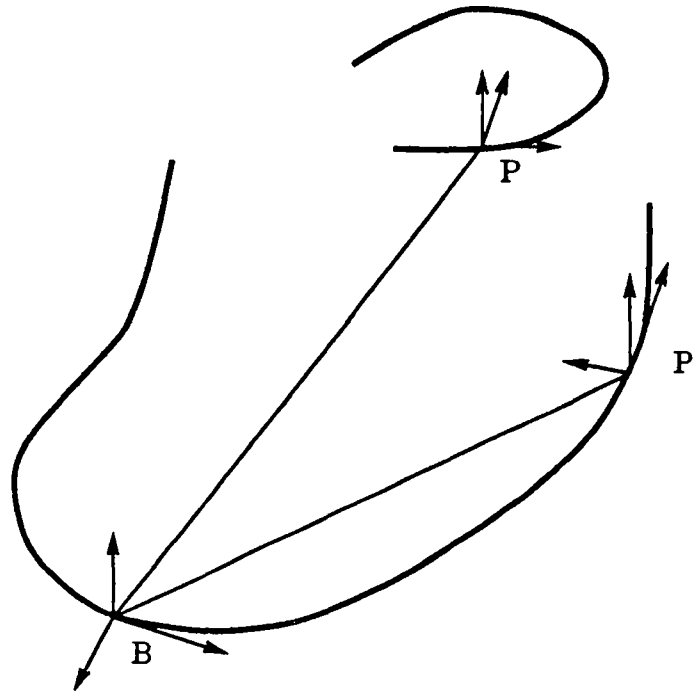

Figure 13: Given a basis point B on the curve, all other points P may be associated with the rigid transformation required to map the Frénet frame at B to the Frénet frame at P. If we also include points P on nearby curves, then the case where a curve is disconnected may still be recognized.

3.2.5 New indexation methods

In the preprocessing phase of the model curves, a basis point B is selected for each such curve, and the $(c, \theta_t, \mathbf{u}_t)$ parameters are calculated for every point P on the (sampled) curve. This computation is repeated for every model curve, *and for every possible basis point* B along the curve.

In this way, the information about the model curves are stored into a three-dimensional table, indexed by $(c, \theta_t, \mathbf{u}_t)$. In each bin of this table, entries consist of a model curve, a point on that curve (together with the corresponding Frénet frames), and a basis point B also on the curve. Each entry in a bin's list contains a model, basis, and point that give rise to the calculation of invarients of the transformation of the basis to the model point that lies in the bin.

For the recognition algorithm, an arbitrary basis point is selected on an unknown curve, and transformations are computed from that basis point to other points along the curve. For each such computation, the parameters $(c, \theta_t, \mathbf{u}_t)$ map to a bin in the three-dimensional table, which gives rise to votes for model/basis pairs, similar to before.

Although this approach has not been thoroughly tested, we expect more stable recognition, based on the statistical studies of the parameters on real data, than with previous indexation schemes. We expect to report on results with this method in the future. We have, however, implemented a recognition system using the curvature and torsion indexa-

tion variables. Experimental results are reported in the next section.

4 Results

Using two views of the skull of Fig. 9, we used existing software to find ridge points, and then fed these points into the curve smoothing algorithm of Section 2. For each view, two non-closed curves are represented: the chin curve, and the orbital ridge (surrounding the eye). Since the two views are of the same skull, there should be a precise correspondence between the two views. Using the indexation algorithm of Section 3.1.3, we preprocess the two curves from one of the views, building the indexation (or hash) table, based on measurements of the curvature and torsion along the points of the curves. Applying the indexation-based recognition algorithm of Section 3.2, the chin curve of the second view of the skull is successfully matched and transformed to the chin curve in the preprocessed view. The resulting match is shown in Fig. 14. Since the output of the matching process includes a rigid transformation, we can apply that transformation to the entire skull, to place the two views in correspondence. The superimposed ridge curves are shown in Fig. 16.

We then apply the matching procedure to the orbital ridge curve of the second view. Although several candidate matches appear, only one of those matches is consistent with the rigid transformation previously identified by matching the chin curves. Incorporating the match of the orbital ridge curves, we can improve the overal rigid transformation estimate, resulting in a more precise correspondence (Figs. 15 and 16).

5 Aknowledgements

We gratefully acknowledge Prof. Robert Hummel, who provided a very important improvement to an original manuscript written in French, and has helped us with very insightful advice and ideas during his stay at Inria. We also thank Serge Benayoun for assistance with the graphical presentation of the results, and the adaptation of the Kalman filter. Finally, we acknowledge the assistance of General Electric-CGR for partial financial support of the research that provided ridge extraction software.

References

[ABB+89] N. Ayache, J.D. Boissonnat, E. Brunet, L. Cohen, J.P. Chièze, B. Geiger, O. Monga, J.M. Rocchisani, and P. Sander. Building highly structured volume representations in 3d medical images. In *Computer Aided Radiology*, 1989. Berlin, West-Germany.

[ABC+90] N. Ayache, J.D. Boissonnat, L. Cohen, B. Geiger, J. Levy-Vehel, O. Monga, and P. Sander. Steps toward the automatic interpretation of 3-d images. In H. Fuchs K. Hohne and S. Pizer, editors, *3D Imaging in Medicine*, pages 107–120. NATO ASI Series, Springer-Verlag, 1990.

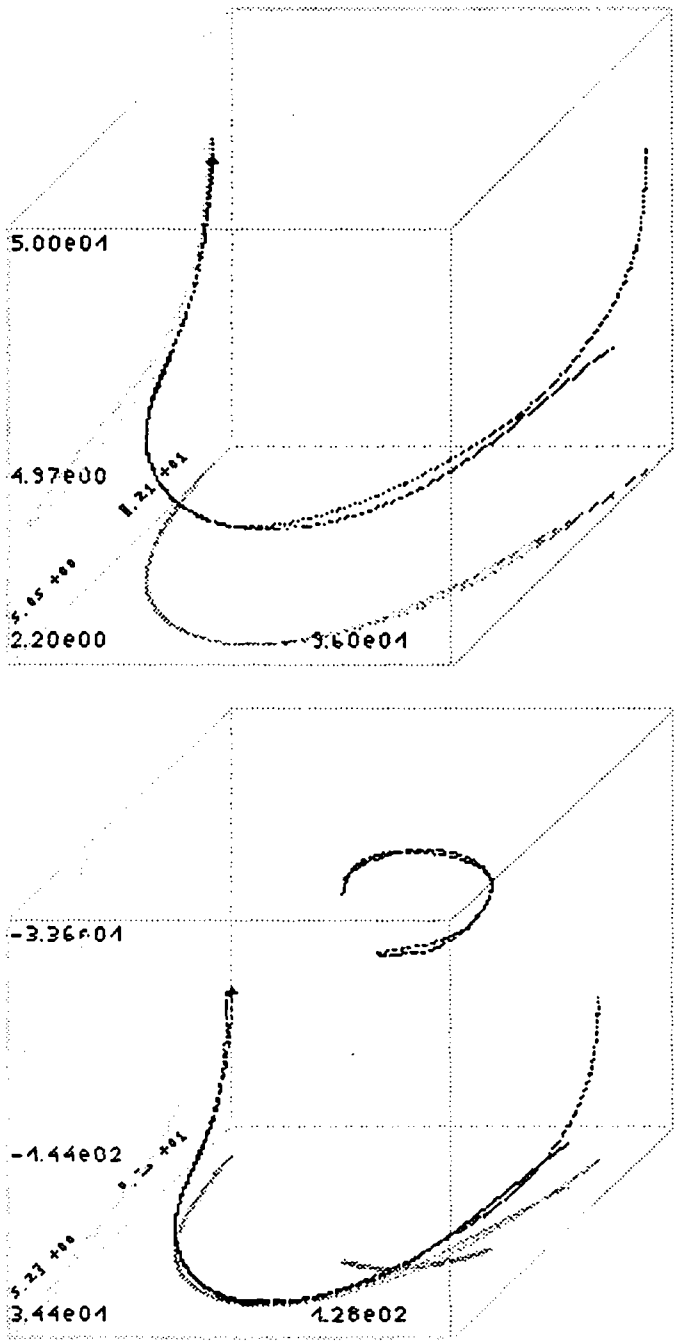

Figure 14: The successful matching of the two chin curves, superimposed. (Note that the occlusion and translation of the second view are handled automatically).

Figure 15: Incorporating the match of the orbital ridge curves improves the global transformation.

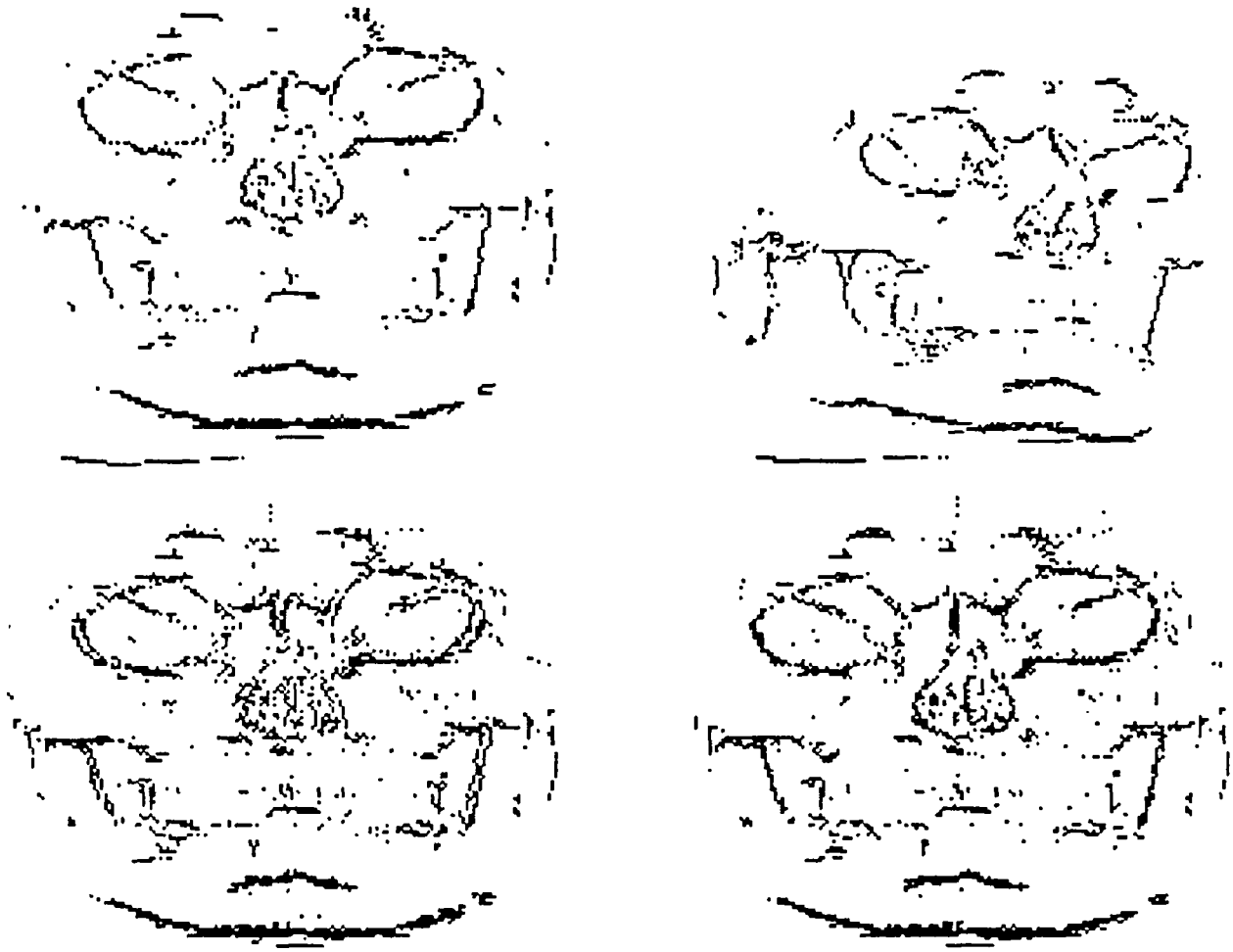

Figure 16: **Registrating the ridges :** top row shows the ridges extracted of a skull scanned in position A (top left) and position B (top right). Figure in the bottom left shows the superposition of the ridge points, obtained after transforming the points of the second view according to the transformation discovered by matching the chin curves. The best correspondences are along the chin points. Figure in the bottom right shows the improved correspondence of ridge points due to the global transformation determined from both matches.

ISSN 0249-6399