International Journal of Computer Vision, 12:1, 79–104 (1994)

Smoothing and Matching of 3-D Space Curves

ANDRÉ GUÉZIEC* AND NICHOLAS AYACHE
INRIA, project Epidaure, F-06902 Sophia-Antipolis
e-mail: Andre.Gueziec@inria.fr and Nicholas.Ayache@inria.fr

Received July 7, 1993.

Abstract

We present a new approach to the problem of matching 3-D 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 Kishon et al. (1990), 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 nonuniform 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 relatively independent 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 3-D medical images. An earlier version of this article was presented at the 2nd European Conference on Computer Vision in Italy.

1 Introduction

Physicians are frequently confronted with the very practical problem of registering 3-D 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, consists of first extracting highly structured descriptions from 3-D images, and then using those descriptions for matching (Ayache et al. 1989; Ayache et al. 1990). The structured descriptions usually come from the extraction of regions of interest, obtained with a 3-D edge extraction algorithm (Monga, Deriche & Rocchisani 1991; Herlin & Ayache 1992), potentially in conjunction with curve or surface deformable models (Cohen & Cohen 1990; Cohen et al. 1991). Then, characteristic curves on the surface are extracted. These curves describe either topological singularities such as surface borders, borders of holes, simple or multiple junctions, etc., (see Malandain et al. (1991)), or

*André Guéziec is currently a Visiting Researcher and an Adj. Faculty at New York University, Courant Institute, 251 Mercer Street, New York, NY 10012.

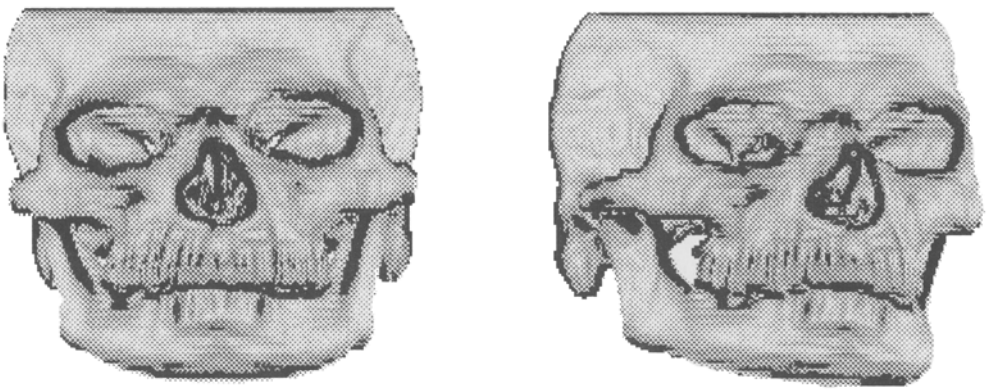


Fig. 1. Extraction of characteristic curves (ridges) from the surface of a skull (using X-ray scanner image data).

differential structures, such as ridges, parabolic lines, and umbilic points (Monga, Ayache, & Sander 1991).

The characteristic curves are stable with respect to rigid transformations, 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 3-D images (involving a few hundreds of points compared to several millions). Figure 1 shows an example of ridges extracted from the surface of a skull (Monga, Benayoun, & Faugeras 1992). 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 (Cutting 1989).

The problem we address in this paper is the use of these curves to identify and accurately locate 3-D 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 intrinsic 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 do Carmo (1976) 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 indexing method, we introduce logical extensions of the work of Kishon et al. (1990), Saint-Marc & Medioni (1990) and Bartels et al. (1987) 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 (Ayache 1986) and (Grimson & Lozano-Peréz 1984) on the identification and positioning of 3-D objects.

We begin by outlining the problems and the approaches.

In section 2, we discuss approaches to fitting curves to collection of voxels (points) in 3-D imagery. Figure 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.

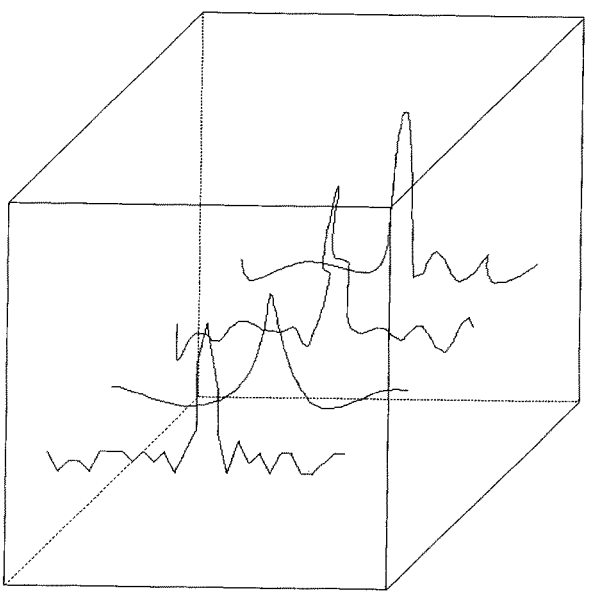


Fig. 2. Given a noisy curve, the degree of smoothing should depend on the noise level, and the desired accuracy in the computation of curvature. All of these curves can be viewed as different noisy versions of a single curve. We applied our smoothing method to curve 1, thus obtaining curves 2, 3, and 4. We notice that the left and right parts of the curve behave differently because the oscillation amplitudes and frequencies are different.

1. We set a threshold ϵ_∞ which is a maximum allowed distance between the approximation 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 3-D image. We then compute a polygonal approximation (of class C^0) of the curve using this threshold (Duda & Hart 1973). Using the vertexes 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 (Pavlidis 1977). 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 penalty 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 indexing 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 indexing-based curve-matching methods.

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

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

Compared with iterative object matching methods, as described by Besl and McKay (1992) and also, for the specific case of curves, as described by Zhang (1992), our method does not need to iterate, and is much faster.

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 estimates 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, discretization, and boundary detection—see Malandain et al. (1991). Especially large deviations must be dealt with as outliers, and filtered before the representation is formed.

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 (1987) is to form the shortest polygonal path through a tube surrounding the polygonal interpolation of the points. Polygonal representations have two major difficulties, however. Vertexes 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 include the class of polygonal curves, offer good approximation properties—see Nilson, Ahlberg, and Walsh (1967)—can readily provide differential information at any point along the spline

curve, and satisfy certain optimality properties, viz., they minimize a certain measure of the bending energy (Holladay 1957). More generally, B-splines of odd degree have best approximation properties, viz. approximation with minimum energy. The reader may consult Nilson et al. (1967). Moreover, successive derivatives are particularly easy to compute by combining lower-order functions. (The formula (1) that we prove later in this section is very similar to the recurrence relation which defines B-spline functions.) There is an extensive literature on B-splines (Bartels et al. 1987; de Boor 1978; Farin 1988; Cinquin 1981, 1987). A survey is provided by Bohm et al. (1984). We provide a very brief introduction, using the notation of Bartels et al. (1987) and Saint-Marc & Medioni 1990).

Given a sequence of $n + 1$ points $P_i(x_i, y_i, z_i)$, $i = 0 \ldots n$ in three-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 $V_j(X_j, Y_j, Z_j)$, $j = 0 \ldots m$ 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 three-space of the approximating curve for a given parameter value $\bar{u}$ is given by

$$Q(\bar{u}) = [X_q(\bar{u}), Y_q(\bar{u}), Z_q(\bar{u})] = \sum_{j=0}^{m} V_j B_{j,K}(\bar{u})$$

$$= \sum_{j=0}^{m} [X_j B_{j,K}(\bar{u}), Y_j B_{j,K}(\bar{u}), Z_j B_{j,K}(\bar{u})]$$

3. The knots must also be specified, and consist of $m + K + 1$ real values $\{\bar{u}_j\}$, with $\bar{u}_0 = 0$ and $\bar{u}_{m+K} = l$, partitioning the interval $[0, l]$ into $m + K$ intervals. This is the de Boor convention. Here, l is the length of the polygon joining the 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:

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

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

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

It is not hard to show that

$$B'_{j,K+1}(\bar{u}) = K \left[\frac{B_{j,K}(\bar{u})}{\bar{u}_{j-K} - \bar{u}_j} - \frac{B_{j+1,K}(\bar{u})}{\bar{u}_{j+K+1} - \bar{u}_{j+1}} \right] \quad (1)$$

We apply the recurrence hypothesis to $B'_{j,K}$ and to $B'_{j+1,K}$ and we notice that

$$\frac{\bar{u}_{j+K+1} - \bar{u}}{\bar{u}_{j+K+1} - \bar{u}_{j+1}} - \frac{\bar{u} - \bar{u}_j}{\bar{u}_{j+K} - \bar{u}_j} = \frac{\bar{u}_{j+K} - \bar{u}}{\bar{u}_{j+K} - \bar{u}_j} - \frac{\bar{u} - \bar{u}_{j+1}}{\bar{u}_{j+K+1} - \bar{u}_{j+1}}$$

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 V_j:

$$\frac{\partial^r Q}{\partial \bar{u}^r} = \sum_{j=0}^{m} V_j B^{(r)}_{j,K}(\bar{u}) = \sum_{j=r}^{m} V_j^r B_{j,K+1-r}(\bar{u}), \qquad V_j^0 = V_j$$

$$V_j^r = (K - r) \frac{V_j^{r-1} - V_{j-1}^{r-1}}{\bar{u}_{j+K-r+1} - \bar{u}_j}$$

Because of this simple formula, we may incorporate constraints 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 $Q(\bar{u})$, the curvature and torsion of the curve at a point $\bar{u}$ are defined by:

$$k(\bar{u}) = \frac{\|Q'(\bar{u}) \wedge Q''(\bar{u})\|}{\|Q'(\bar{u})\|^3},$$

$$\tau(\bar{u}) = \frac{\det [Q'(\bar{u}), Q''(\bar{u}), Q'''(\bar{u})]}{\|Q'(\bar{u}) \wedge Q''(\bar{u})\|^2}$$

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

$$\mathbf{t}(\bar{u}) = \frac{Q'(\bar{u})}{\|Q'(\bar{u})\|},$$

$$\mathbf{b}(\bar{u}) = \frac{Q'(\bar{u}) \wedge Q''(\bar{u})}{k(\bar{u})\|Q'(\bar{u})\|^3}$$

and **n** is defined as the vector cross product of **b** and **t**.

2.2 *A Previous Approximation Scheme*

We next recall a classic approximation scheme due to Barsky (Bartels et al. 1987). This scheme has been used by Saint-Marc and Médioni (1990) for curve matching. Our emphasis is on the shortcomings of the approach of 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, perturned by identically distributed Gaussian noise. A least-squares derivation then corresponds to a maximum-likelihood reconstruction of the data (Lancaster & Salkauskas 1986).

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

$$\Delta_1 = \sum_{i=0}^{n} \|Q(\tilde{u}_i) - P_i\|^2$$

$$= \sum_{i=0}^{n} \{[X_q(\tilde{u}_i) - x_i]^2 + [Y_q(\tilde{u}_i) - y_i]^2 + [Z_q(\tilde{u}_i) - z_i]^2\}$$

$$= \sum_{i=0}^{n} \left\{ \left[\sum_{j=0}^{m} X_j B_{j,K}(\tilde{u}_i) - x_i \right]^2 + \left[\sum_{j=0}^{m} Y_j B_{j,K}(\tilde{u}_i) - y_i \right]^2 + \left[\sum_{j=0}^{m} Z_j B_{j,K}(\tilde{u}_i) - z_i \right]^2 \right\}$$

The calculation of the $\tilde{u}_i$ values is critical, since $\|Q(\tilde{u}_i) - P_i\|$ is supposed to represent the Euclidian distance of the point 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 $Q(\bar{u})$. As an expedient, Barsky suggests using for $\tilde{u}_i$ the current total length of the polygonal curve from P_0 to P_i, realling that $0 \leq \bar{u} \leq L$, and L is the length of the entire polygonal path (see figure 4). Thus as an estimate, we can use

$$\tilde{u}_p = \sum_{i=0}^{p-1} \|\mathbf{P}_{i+1} - \mathbf{P}_i\| \tag{2}$$

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

$$\frac{\partial \Delta_1}{\partial X_j} = 2 \sum_{i=0}^{n} B_{j,K}(\tilde{u}_i) \left[\sum_{l=0}^{m} X_l B_{l,K}(\tilde{u}_i) - x_i \right] = 0$$

$$\frac{\partial \Delta_1}{\partial Y_j} = 2 \sum_{i=0}^{n} B_{j,K}(\tilde{u}_i) \left[\sum_{l=0}^{m} Y_l B_{l,K}(\tilde{u}_i) - y_i \right] = 0$$

$$\frac{\partial \Delta_1}{\partial Z_j} = 2 \sum_{i=0}^{n} B_{j,K}(\tilde{u}_i) \left[\sum_{l=0}^{m} Z_l B_{l,K}(\tilde{u}_i) - z_i \right] = 0$$

As j ranges from 0 to m, we obtain three 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

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

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

$$AX = B_x, \quad AY = B_y, \quad AZ = B_z \tag{3}$$

we discover that A is a symmetric banded matrix with $2K - 3$ nonzero bands (for a 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 vertexes. The system then has the form

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

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 Cholesky decomposition. Instabilities of the spline fit can appear in the case when a model has a larger number of control vertices $m + 1$ than needed (e.g., consists of a straight line). This is due to the fact that splines are not well conditioned and a small perturbation of the tangent at the origin will produce oscillations in the whole curve. Numerical problems occur if we interpolate the data rather than approximate. This is due to Cholesky's method. We have recently discovered that efficient methods alleviate this problem—see de Hoog & Hutchinson (1987)—via a QR-factorization of a different matrix.

There is an easier way to obtain the linear system (3) if we use a matrix notation: We call B the matrix of the $B_{j,K}(\tilde{u}_i)$ with $m + 1$ lines and $n + 1$ columns, X the matrix of the control vertices with $m + 1$ lines and three columns, x the $n + 1$ by three matrix of data points coordinates: $\Delta_1 = (B^tX - x, B^tX - x)$, (x, y) is the scalar product of vectors x and y. By differentiating the scalar Δ_1 with respect to the matrix X, one obtains

$$BB^tX - Bx = 0, \quad \text{thus} \quad AX = Bx$$

From the equality $X^tBB^tX = \|B^tX\|^2$, we know that A and B have the same rank. Thus if $m \leq n$, A is symmetric positive definite up to numerical error as well as the inverse A^{-1} ($AA^{-1} = I = (A^{-1})^t A^t = (A^{-1})^t A$). The B-spline solution can be written as $Sx = B^t A^{-1} Bx$.

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 (figure 3a), we reconstruct figure 3b 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 figures 3c and 3d 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 (Pavlidis 1977). We make use of this property to distribute B-spline knots nonuniformly 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

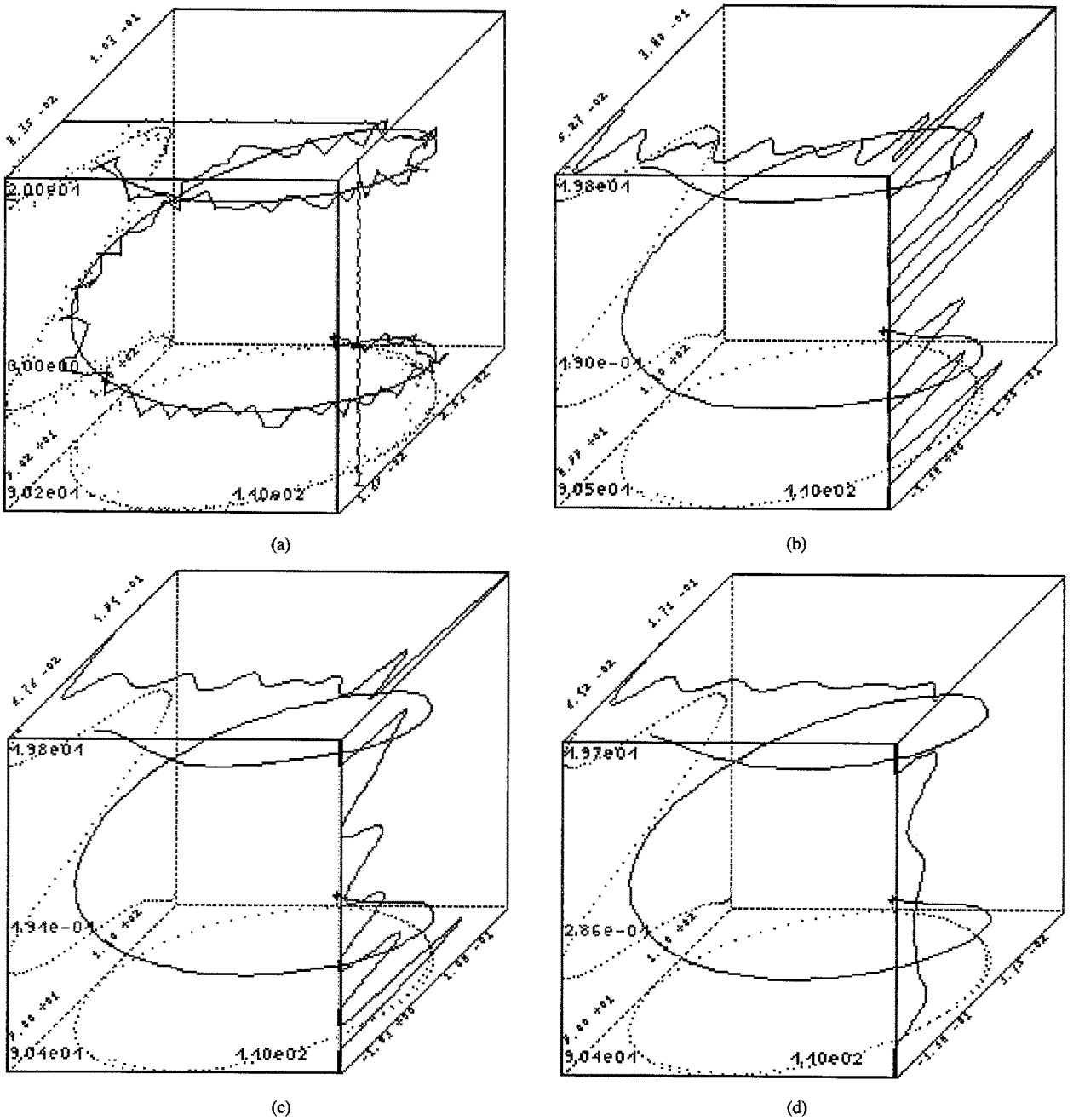


Fig. 3. (a) Noise is added to a helix, and points are sampled with a limitation on the distance between successive points. The curvature and torsion are plotted in the top and the right panels of the cube, as a function of arclength. In a perfect reconstruction, the curvature and torsion would be constant. (b) 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. (c) A more precise estimate of model-data distances improves the estimation of curvature and torsion. (d) The constraint on the second derivative also improves the estimation. In fact the torsion should not be better estimated, but for helices the product curavature-torsion is a constant.

portions of the curve. In order to obtain the initial polygonal approximation, we use the classical recursive algorithm of Duda and Hart (1973), 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 *IP* (see A in figure 4), as opposed to the approximating polygonal path *AP* (see B in figure 4), we simply project each point P_i onto the approximating polygonal path *AP* and considering the relative position of the projected points in terms of total cordlength of the path. This method, projection onto the approximating polygonal path, seems to outperform Barsky's proposal, especially when data points are densely populated around the curves. To set the knot values of the spline ($m + K$ real values $\{\bar{u}_j\}$), we notice that for a B-spline of order 1, that is a polygon, approximating the data within the tolerance ϵ_∞, the best $\bar{u}_j$ correspond to cordlength values on the approximating polygon *PA*. In this case, the global parameter $\bar{u}$ is the cordlength and the best order 1 spline is this very polygon *PA*. We simply keep the same knot values for higher-order splines. Guided by texts such as Marin (Marin 1984), we are now investigating possibly better knot values for higher-order splines. We might optimize $\{\bar{u}_j\}$ values with a gradient descent approach as Wang and Ferrari do in Irvine (Wang 1990). But our solution seems to be a good compromise, as our experimental results show.

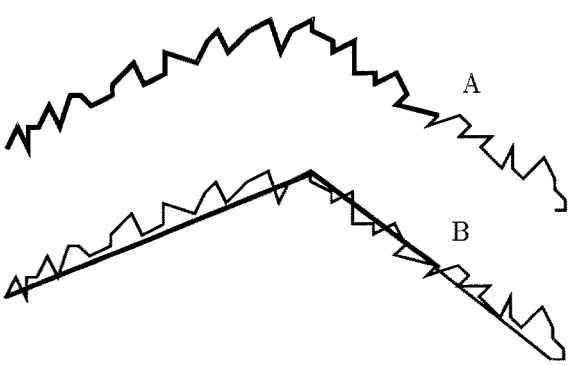


Fig. 4. A comparison of the initial positions for the projected point locations $\tilde{u}_i$, represented as distances (thick line) on a polygon using Barsky's method or interpolating polygon *IP* (A) and our modification or approximating polygon *AP* (B). The tolerance ϵ_∞ of this polygon fit is related to the quality of the data and the discussion in section 1.

2.3.2 Improved Distance Estimates (Plass & Stone 1983). Estimation of the $\tilde{u}_i$ 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 $\|Q(\tilde{u}_i) - P_i\|$. Let us thus consider the following equation, where $\tilde{u}_i$ is unknown:

$$F_i(\tilde{u}_i) = \frac{\partial \|Q(\tilde{u}_i) - P_i\|}{\partial \tilde{u}_i} = 0$$

$$= \frac{\partial \|\Sigma_{j=0}^{m} V_j B_{j,K}(\tilde{u}_i) - P_i\|}{\partial \tilde{u}_i}$$

$$= 2 \left[\sum_{j=1}^{m} \frac{(K-1)(V_j - V_{j-1})}{\bar{u}_{j+K-1} - \bar{u}_j} B_{j,K-1}(\tilde{u}_i) \right] \cdot \left[\sum_{j=0}^{m} V_j B_{j,K}(\tilde{u}_i) - P_i \right]$$

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

$$DV_j = \frac{(K-1)(V_j - V_{j-1})}{\bar{u}_{j+K-1} - \bar{u}_j}$$

$$D^2V_j = \frac{(K-2)(DV_j - DV_{j-1})}{\bar{u}_{j+K-2} - \bar{u}_j}$$

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

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

Despite the apparent complexity of these equations, these computations are not very expensive, since $B_{j,K-1}$ and $B_{j,K-2}$ 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 vertexes $\{V_j\}$. We may repeat this operation until the solution satisfies the ϵ_∞ criterion or until no substantial decrease of the maximum error is observed. Figure 5 illustrates our method. Moreover, for security, $\{\tilde{u}\}$ values must be sorted and rescaled after each iteration. There is no guaranty for the convergence for this method, neither locally for a specific value $\tilde{u}_i$ nor a fortiori globally for all $\{\tilde{u}\}$. The reader will find in (Grossman 1971) a description of the problems that could arise. Thus we do a divergence test and stop the process if we encounter a bad case. Figure 3c shows the improvement that iterations effect on the approximation curve.

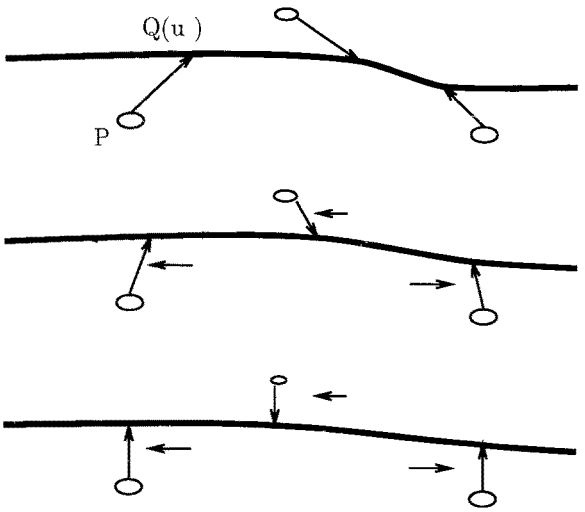


Fig. 5. For each point P, the corresponding $\tilde{u}$ value follows a gradient descent. The least square curve is modified each time the $\{\tilde{u}\}$ values are modified.

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 Minimization of Curvature. Cubic B-splines have an optimal approximation property (Nilson et al. 1967), namely, that among all interpolants they minimize the norm of the second derivative. This result was first proven by Holladay (1957). 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:

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

where

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

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, with a smoothing parameter τ set as $\tau = (\sigma_2/\sigma_1)^2$. In appendix A we discuss several ways to optimize τ. But it is NOT the bending energy. We have here a compromise that provides a better curvature evaluation with a small additional computation. Note that with our implementation, the sum can be taken over more than $n + 1$ points, thus accounting for high curvature sections of the curve between sample points. Since the second derivative values are linear in terms of control vertices, Δ_2 is again quadratic. With the previous matrix notations, and with B'' as the $m + 1 \times n + 1$ matrix of the $B''_{j,K}(\tilde{u}_i)$ we can write

$$\frac{\Delta_2}{\sigma_1^2} = (B^tX - x,\ B^tX - x) + \tau(B''X,\ B''X)$$

$$A_\tau = BB^t + \tau B''B''^t$$

We solve a linear system of size $m + 1$, and the construction and complexity are as before, and the result is a spline $Sx = B^t A_\tau^{-1} Bx$. 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. Figure 3d illustrates results of minimizing Δ_2.

We have also tried to constrain the third derivative for cubic splines or the fourth derivative for quartic splines to improve the estimation of the torsion. This idea of minimizing the highest nonzero derivative is not so classical as we thought. However, we discovered that Arbogast (1990) experienced it. This derivative is piecewise constant for splines and presents jumps. These jumps can be used as a measure of the irregularity of the spline. By penalizing higher derivatives, we smooth them out.

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 $\mathbf{n_i}$. Monga, Ayache, and Sander (1991) study the stability of surface normal measurements. Accordingly, we penalize our optimization criterion by a measure of the violations of this condition:

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

with

$$\sigma_3^2 = \text{var}\ [Q'(\tilde{u}_i) \cdot \mathbf{n_i}]$$

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 3-D 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 3-D point. Also, note that although the surface normal $\mathbf{n_i}$ is *not* in general identical to the curve normal n (see figure 6), it is always orthogonal to the curve tangent Q', establishing the validity of our Δ_3 criterion.

We note N, the $n + 4$ matrix of normal vectors $\mathbf{n_i}$ associated to curve points gathered in matrix x, and ν the regularization parameter related to the surface normal penalization term ($\nu = (\sigma_3/\sigma_1)^2$). Moreover, we introduce the following $3(m + 1) \times 3(n + 1)$ matrixes:

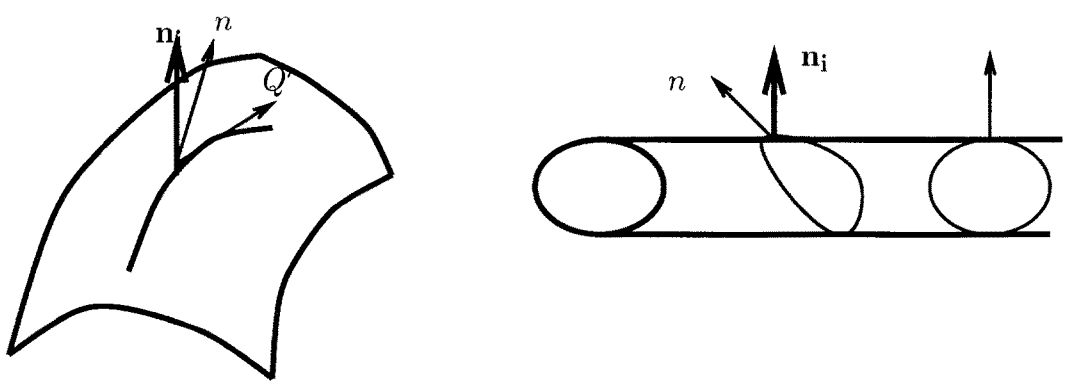


Fig. 6. The curve tangent lies in the tangent plane of the surface and is thus orthogonal to the surface normal.

$$N_3 = [\text{diag } (Nx) \text{ diag } (Ny) \text{ diag } (Nz)]$$

$$B_3 = \begin{pmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{pmatrix}$$

$$B_3' = \begin{pmatrix} B' & 0 & 0 \\ 0 & B' & 0 \\ 0 & 0 & B' \end{pmatrix}, \text{ etc.}$$

With the new vectors X_3—the $3(m + 1)$ vector representing X-in-fact for the machine, X_3 and X are identical—and x_3—the $3(n + 1)$ vector representing x—we can write

$$\frac{\Delta_3}{\sigma_1^2} = (B_3^t X_3 - x_3,\ B_3^t X_3 - x_3) + \tau(B_3''^t X_3,\ B_3''^t X_3) + \nu(N_3^t B_3''^t X_3,\ N_3^t B_3''^t X_3)$$

and thus

$$A_{3\tau,\nu} = B_3 B_3^t + \tau B_3'' B_3''^t + \nu B_3' N_3 N_3^t B_3''^t$$

Finally, Δ_3 is still quadratic, the result is again a spline $Sx_3 = B_3^t A_{3\tau,\nu}^{-1} B_3 x_3$ but due to the scalar product, variables cannot be separated and the system size is multiplied by three. The matrix $A_{3\tau,\nu}$ is no more a band matrix and the complexity of the resolution is in $O(m^3)$.

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 (crest lines). In figure 13 we will show such crest lines on the same skull scanned in to different positions A and B, which permit a matching accuracy of one voxel (our modelization of the chin curve is represented in figure 7a). The two crest lines are also present in view B, with an occlusion and a small shift of the (perceptual) origin for the chin B. There are no obvious matching pairs of points between the two views (see figure 7b). Using surface normals in figure 7c, the curvature values of the chin curve in view B are closer to those on the first view A. 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. However, it is possible to compute useful torsion measures when we penalize a higher-order derivative than the second-order derivative. We did some promising tests combined with the automatic determination of the regularization parameter, these are summarized in appendix A (see especially figure 18).

In summary, smoothing with regularized B-splines can be done in linear time in terms of data points and cubic time in terms of control vertices, so in $O(n + m^3)$. 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: (i) identify a curve M_i which has the largest subset of points in common with S after a rigid transformation; and (ii) 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 number 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 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 that brings the two frames into correspondence is given simply by the outer product:

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

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

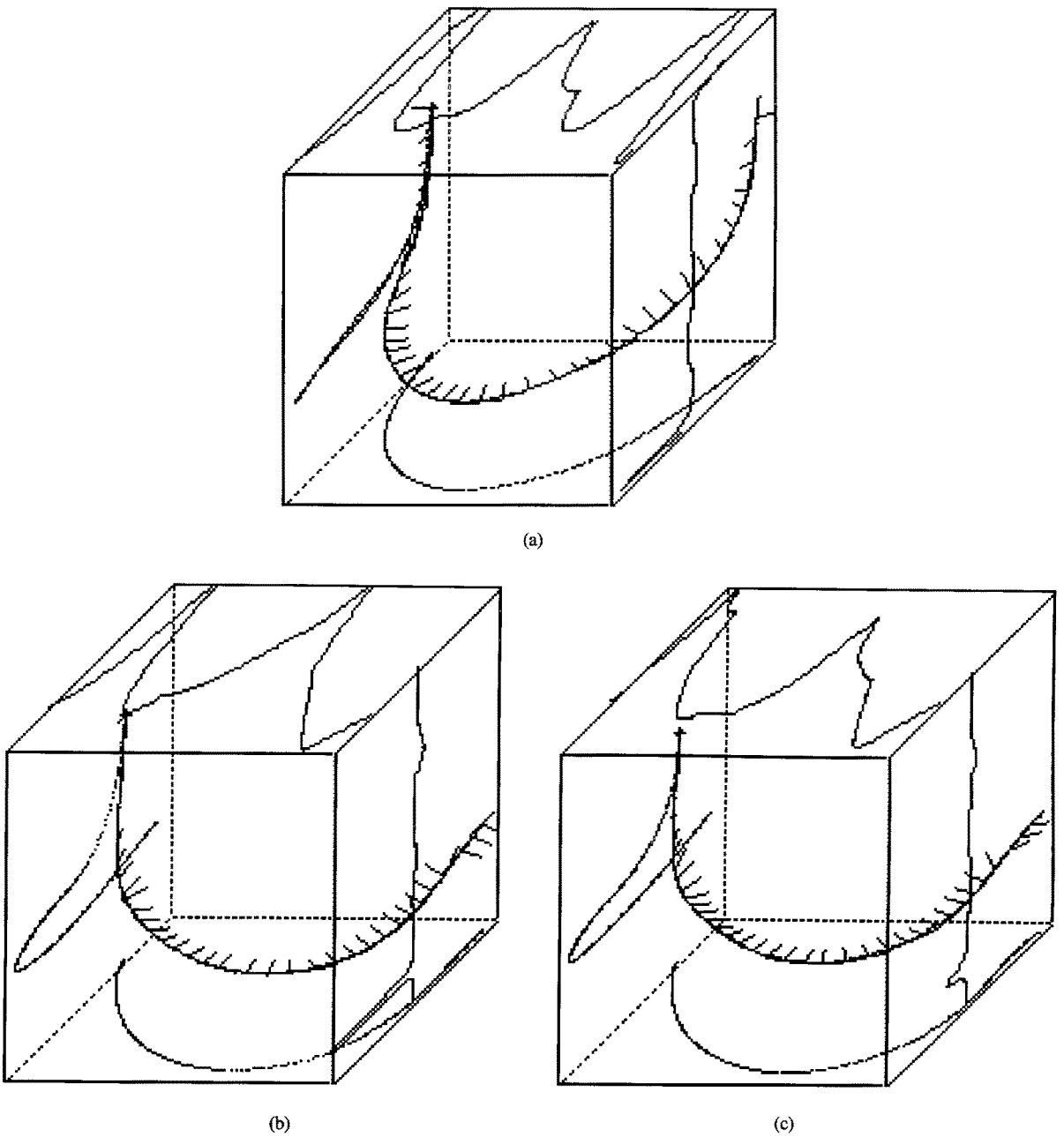


Fig. 7. Approximation of the chin A by our method. We observe curvature (top), and torsion (right) in terms of arclength, as well as curve tangent and normal along the curve. All improvements of section 2.3 have been implemented for this example. (b) The chin on view B is occluded. We do not use here the surface normal penalization. The curvature function is slightly different from the view A. (c) With the help of surface normals, the curvature representation is much more reliable (the centered double peak is in A and B).

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 or Alignment. 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 refined transformation may be computed using a least-mean-squares approach, iteratively improving the hypothesis, for example by making use of a Kalman filter (Ayache 1991).

By checking on all possible hypotheses, we find the pair that 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 $O(kn^2 \log n)$. This 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 any one 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 $O(n \log n)$ and $O(n^2 \log n)$, 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 *indexing 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 model 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. This is *not* a Hough transform. Instead of accumulating over model-scene pairs of points we simply process the scene points *once*, provided preprocessing of the models be done off-line.

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 complexity lies somewhere between $O(n)$ and $O(mn^2)$, depending on the level of quantization of the index space according to curvature and torsion. The length of a bin's entry is at the worst $O(mn)$. 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 are constant along a helix.

This description of the method of indexing is essentially the "geometric hashing" method of Kishon, Hastie, and Wolfson (1990), 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 parameterization 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 that 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 a uniform (approximate) rigid transformation. (Our approach, while different, may also be compared with that of Stein (1991).)

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 indexing 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. The results of the complexity study for the three different models (Alignment, Accumulation, Indexing) are summarized in the accompanying table, where m is the number of models:

complexity/ method	alignment	accumulation	indexing
lower bound	$O(mkn \log n)$	$O(mn \log n)$	$O(n)$
upper bound	$O(mkn^2 \log n)$	$O(mn^2 \log n)$	$O(mn^2)$

3.2 Enhancements to the Indexing Method

Since the indexing 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 indexing method we have employed.

3.2.1 Indexing Table Quantization. Guided by Grimson and Huttenlocher (1991), 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 intercurve 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. We then trace an uncertainty ellipse around curvature and torsion measures. The principal axes of this ellipse are given by intracurve variances. We add an entry in hash-table to each cell that has a nonempty intersection with the ellipse.

3.2.2 A Metric for Rigid Transformations. At the same time, we compute covariance values for the six-parameter rigid transformations 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 indexing algorithm.

3.2.3 Recursive Transformation Estimation. Throughout the recognition phase, as soon as a pair of points is 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 figure 8). For each point P, we can define the rigid transformation $D = (r, \mathbf{u})$ that maps the Frénet frame at B onto the Frénet

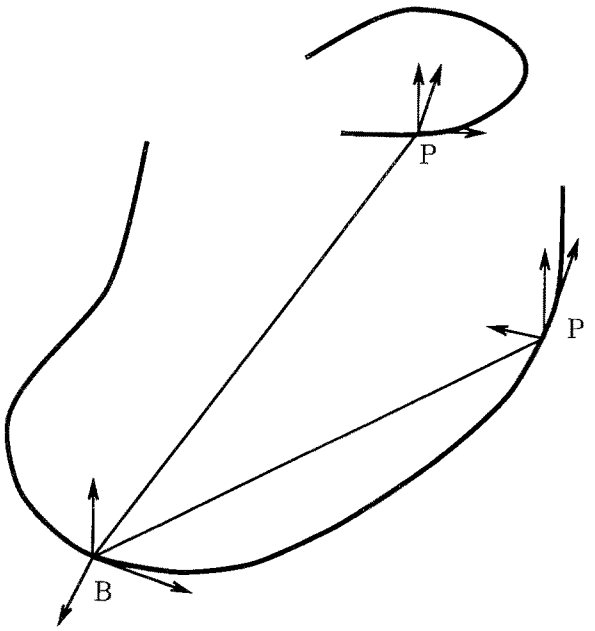


Fig. 8. Given a basis point B on the curve, all other points P may be associated with the rigid transformation required to map the Frénet frame at B to the Frénet frame at P. If we also include points P on nearby curves, then the case where a curve is disconnected may still be recognized.

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, and the translation coordinates $(\mathbf{u}_t, \mathbf{u}_n, \mathbf{u}_b)$, again measured in the basis frame.[1] 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 sampled each spline curve with a uniform sampling. A single corresponding basis point B 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 intercurve variances are computed from the set of all pairs of corresponding points P on different curves, and the intracurve variances variances are computed using distinct points P along a single curve (see figure 9). Variances are computed for every invariant. The quotient between the intercurve variance and the intracurve variance measures the stability of the corresponding invariant. The results of this study are summarized in the accompanying tabulation.

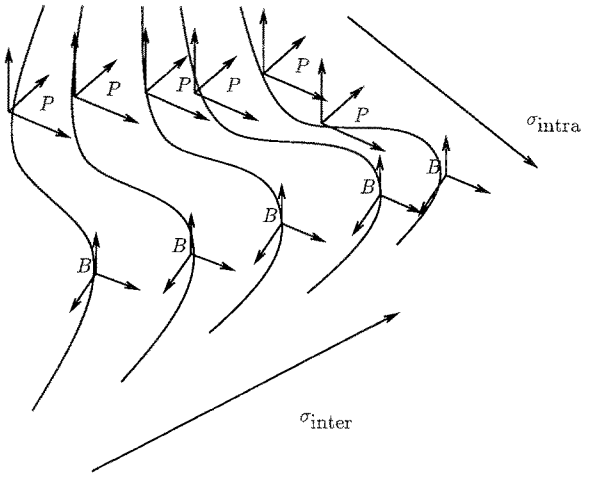


Fig. 9. Statistical Experiments: A single corresponding basis point B is identified on all of the curves, and other homologous points P may be determined from the sample number. The intercurve variances are computed from the set of all pairs of corresponding points P on different curves, and the intracurve variances are computed using distinct points P along a single curve.

σ	c	τ	$\|\mathbf{u}\|$
inter	0.0034	0.4069	0.8932
intra	0.0192	0.5137	31.774
ratio	**0.17771**	*0.7921*	0.0281
σ	θ_t	θ_n	θ_b
inter	0.0368	0.0832	0.0978
intra	0.8822	0.5479	0.6602
ratio	**0.0417**	0.1518	0.1477
σ	$\mathbf{u}_t$	$\mathbf{u}_n$	$\mathbf{u}_b$
inter	0.8267	1.0089	2.4657
intra	36.350	8.8191	38.836
ratio	**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 indexing algorithm that we describe in the next subsection.

3.2.5 New Indexing Method. 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, information about the model curves is 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 invariants 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. This procedure applies also to curves in *multiple sections* (features are exclusively local), and last, since our invariant measures are local, to *scattered points* associated with curvature information and a local reference frame. Experimental results are reported in the next section. In appendix B we discuss a Bayesian analysis of this matching method done by Rigoutsos and Hummel (1991). It permits a likelihood value to be associated to each match. It also explains how we can refine a local match and obtain a global match by taking more curves into account, as will be illustrated in figure 12.

4 Results

4.1 Matching Two Pairs of curves

Using two views of the skull of figure 1 and figure 11a, we used the software of Monga, Benayoun, and Faugeras (1992) to find points along crest lines, and then fed these points into the curve-smoothing algorithm of section 2. For each view, two nonclosed 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 indexing algorithm, we preprocess the two curves from one of the views, building the indexing (or hash) table, based on measurements of $(c, \theta_t, \mathbf{u}_t)$ along the points of the curves. Applying the indexing-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 figure 10a. 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 figure 11c.

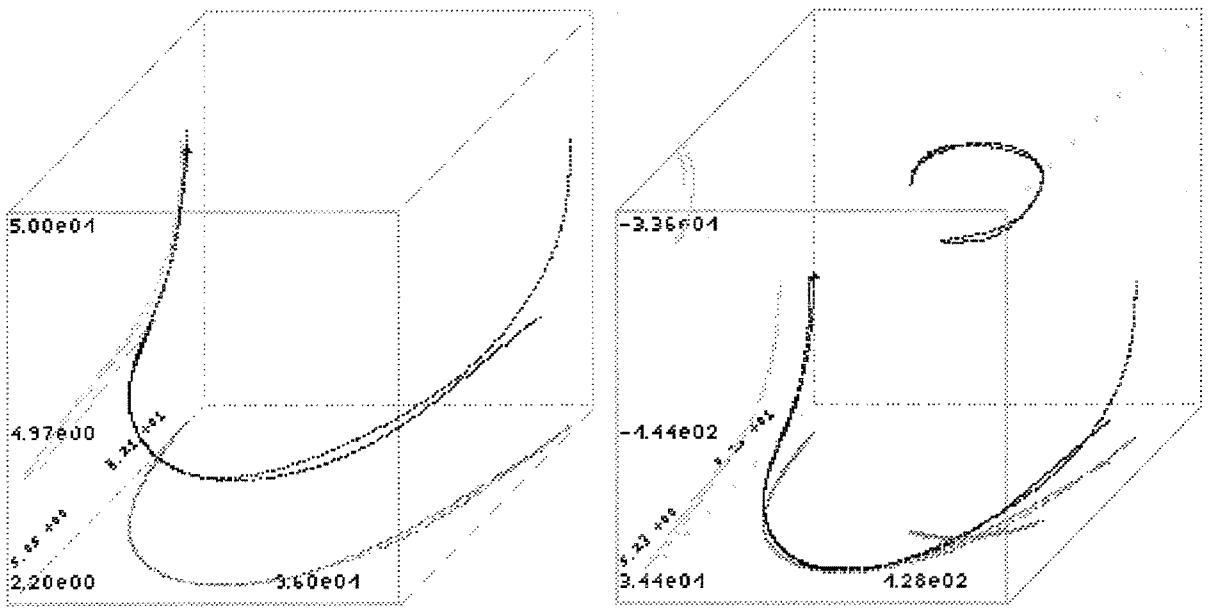
We then apply the matching procedure to the orbital ridge curve of the second view. Although several candidate matches appear, only one of those matches is consistent with the rigid transformation previously identified by matching the chin curves. Incorporating the match of the orbital ridge curves, we can improve the overall rigid transformation estimate, resulting in a more precise correspondence (figures 10b and 11d).

4.2 Matching Simultaneously Several Curves and Scattered Points.

We preprocessed all curves from A in this experiment, and applied the algorithm of section 3.2.5. Curves from B were successfully identified. The resulting transformations were applied to all curves from B and superimposed matches (model, scene) appear in figures 12b and 12d. We next ran our algorithm on the chin and right orbit curves considered as one single curve (figure 12e) and finally on all curves simultaneously (figure 12f). CPU times on a DEC-workstation (in seconds) for recognition and positioning are summarized in the following table. It confirms the linear time hypothesis.

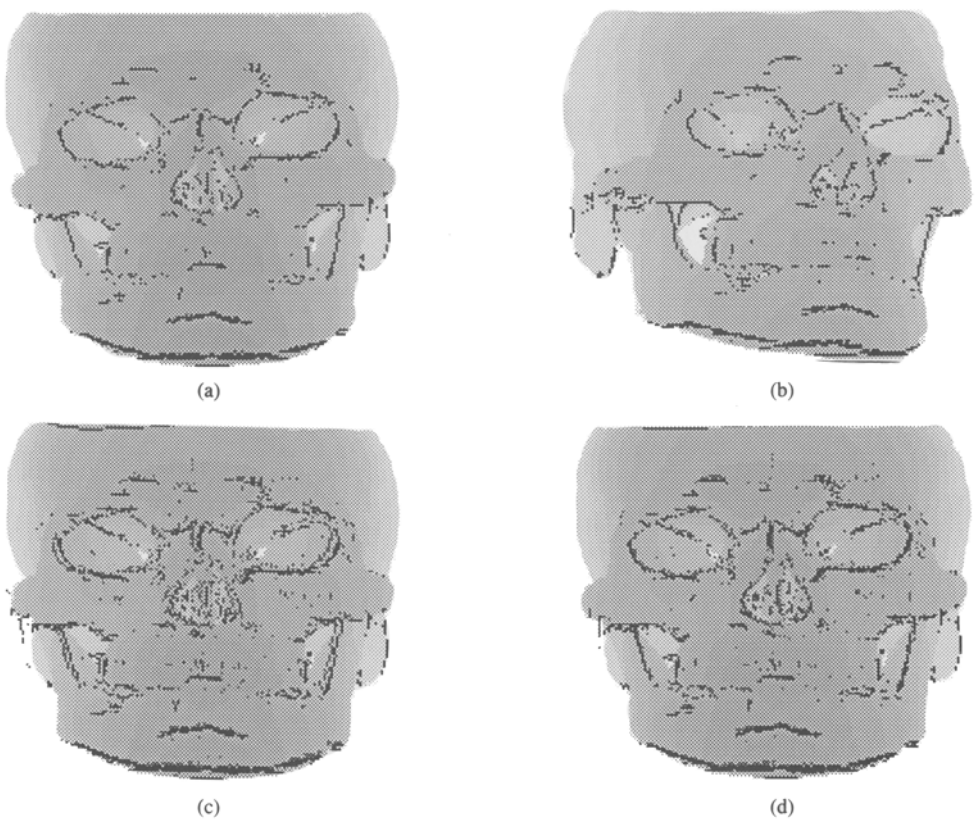
scene curve	noise contour	right orbit	left orbit	chin	chin orbit	all curves from B
CPU time	1.08	0.96	1.18	2.57	3.56	9.51

Note that incorporating more curves increased the *likelihood* of the match (see appendix B). We thus start from a local curve match and end up with one global rigid transformation. We then experimented matching by using scattered points (several hundreds) on the surface of the object, selected for the high curvature value on the surface and associated with a surface frame (Monga, Benayoun & Faugeras 1992) (figure 12g). Last, we registered the entire skull by just applying the



(a) (b)

Fig. 10. (a) The successful matching of the two chin curves, superimposed. (Note that the occlusion and translation of the second view are handled automatically). (b) Incorporating the matching of the orbital ridge curves improves the global transformation.



(a) (b)

(c) (d)

Fig. 11. (a) The extracted ridges of a skull scanned in position A and (b) in position B. (c) 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. (d) The improved correspondence of ridge points due to the global transformation determined from both matches.

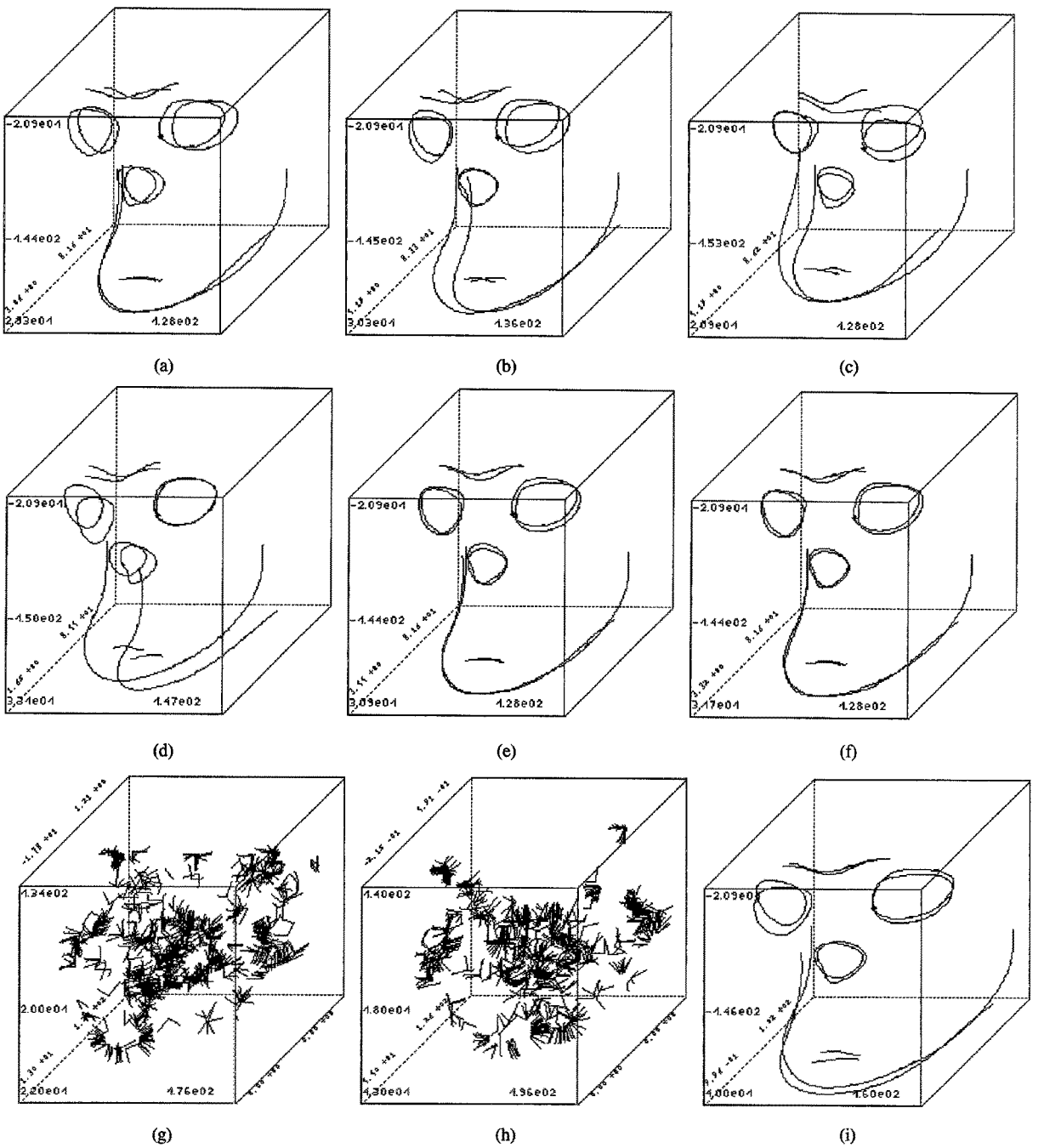


Fig. 12. (a) The successful matching of the two-submandibular curves, superimposed. (Note that the occlusion and translation of the second view are handled automatically.). (b) Nose contours matched. (c) Right orbits matched. (d) Left orbits matched. (e) Chin-orbit matched simultaneously. (f) All curves matched simultaneously. The matching algorithm is successfully applied in (i) to scattered points associated to (g) A and (h) B, represented here together with their reference frame. There is a scale factor on the x and y axes, due to the evaluation in image coordinates (as compared to real coordinates in the previous plots).

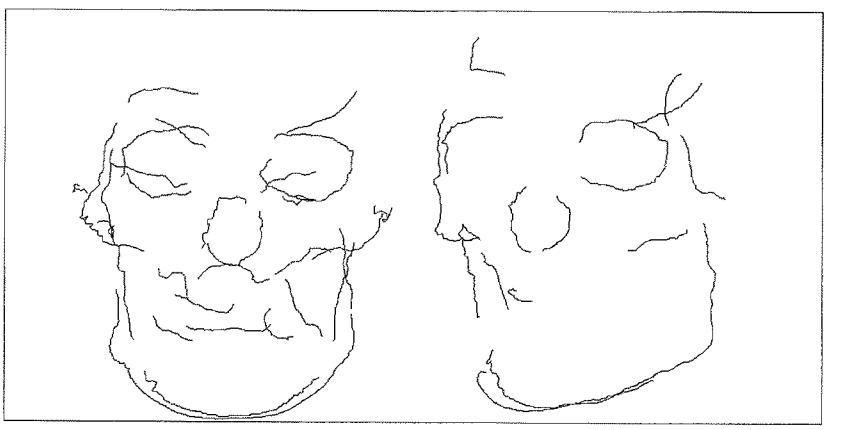
transformation that superimposed the two submandibular curves. Incorporating the match of the orbital ridge curves, we improved the overall rigid transformation estimate, resulting in a more precise correspondence (figure 11).

4.3 Matching Dozens of Curves and Thousands of Points in a Few Seconds

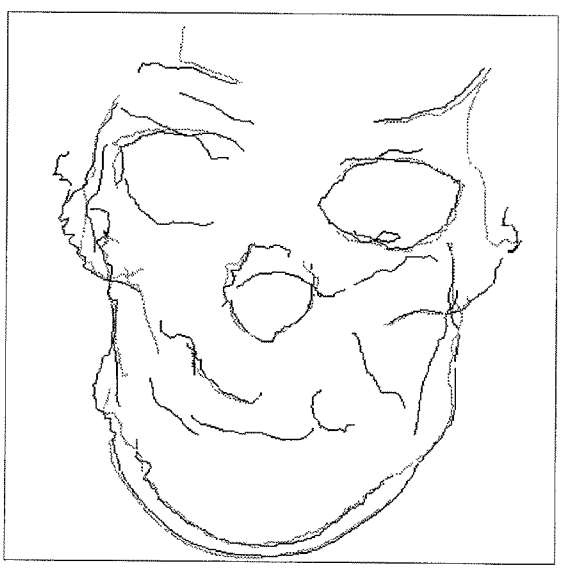
We present the latest matching of crest lines extracted with the algorithm of Thirion and Gourdon (1992). Figure 13 shows lines from our skull scanned in two different positions, as well as the superposition of those lines with our algorithm. These dozens of lines feature thousands of points in total. The matching takes a few seconds on a DEC 5900.

Figure 14 shows two CT-scan images of a vertebra. Figure 15 shows crest lines obtained from the two different views of the vertebra of figure 14, as well as the matching of the crest lines.

In figure 16 we display new CT-scan data of the head. Figure 17 shows crest lines obtained on data from figure 16 along with the registration of the crest lines.



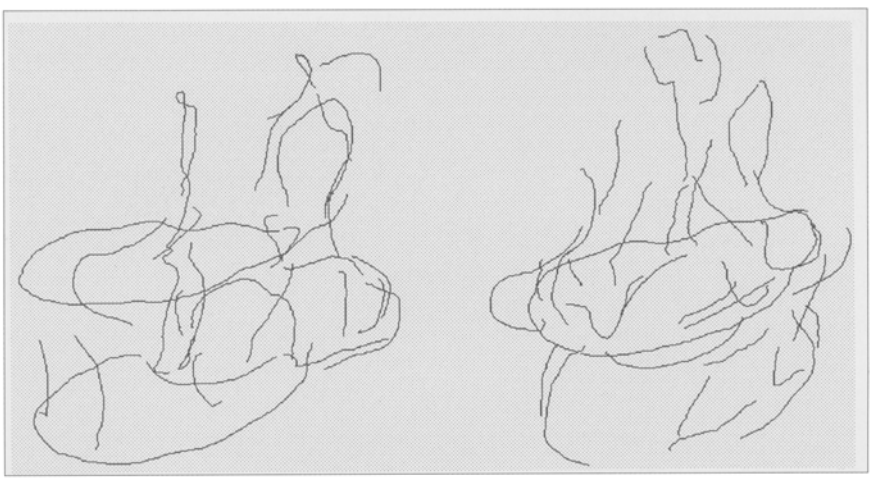
(a)



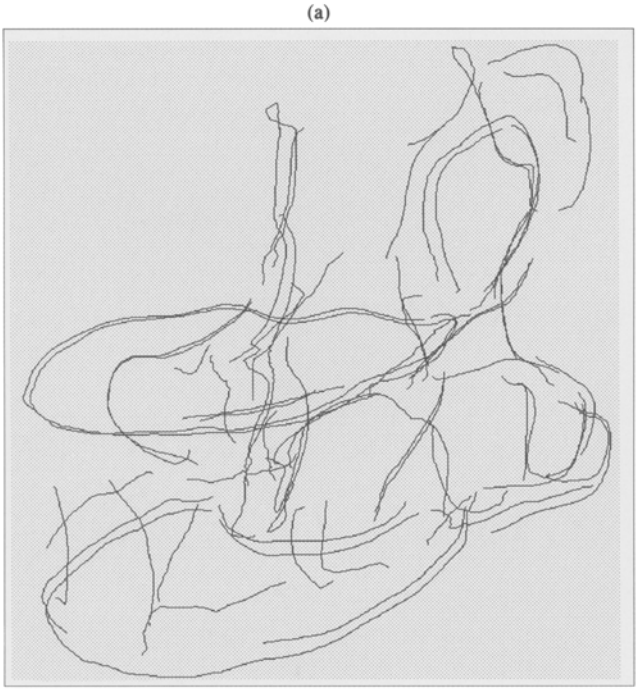
(b)

Fig. 13. (a) Crest lines on two images of a skull obtained with the algorithm of Thirion and Gourdon (1992). (b) Superposition of the lines of (a) with our software in 7.71 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.

Fig. 14. Isosurfaces on two CT-scan images of a vertebra (courtesy of J.L. Coatrieux and R. Collorec).



(a)



(b)

Fig. 15. (a) Crest lines on two images of a vertebra obtained with the algorithm of Thirion and Courdon (courtesy of J.L. Coatrieux and R. Collorec). (b) Superposition of the lines of (a) and our software in 8.85 seconds CPU on a DEC 5900. Twelve curves were identified out of 28 and 33. Eighty-two points were identified among these curves out of 1250 and 1213 points respectively in the two different sets of curves.

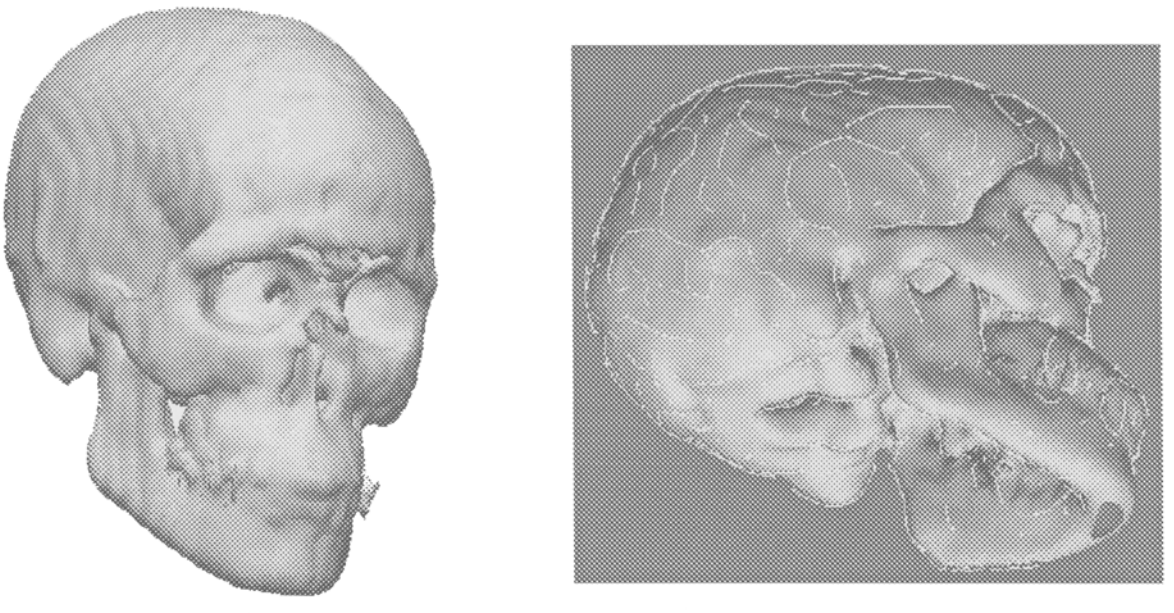
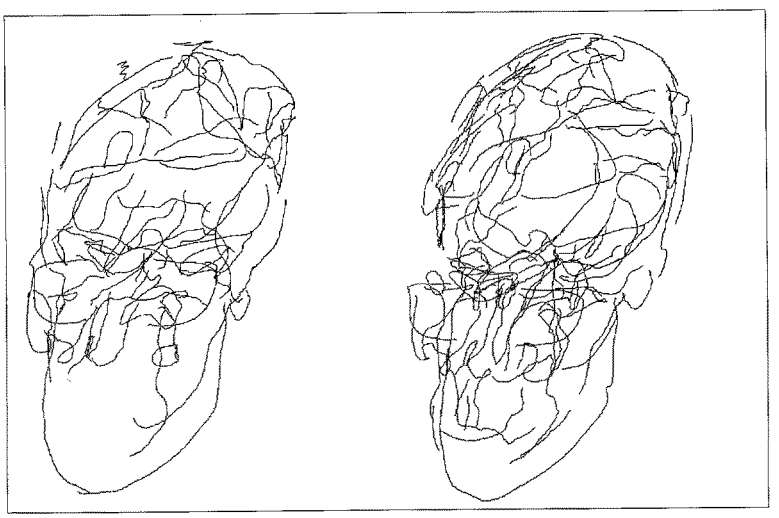
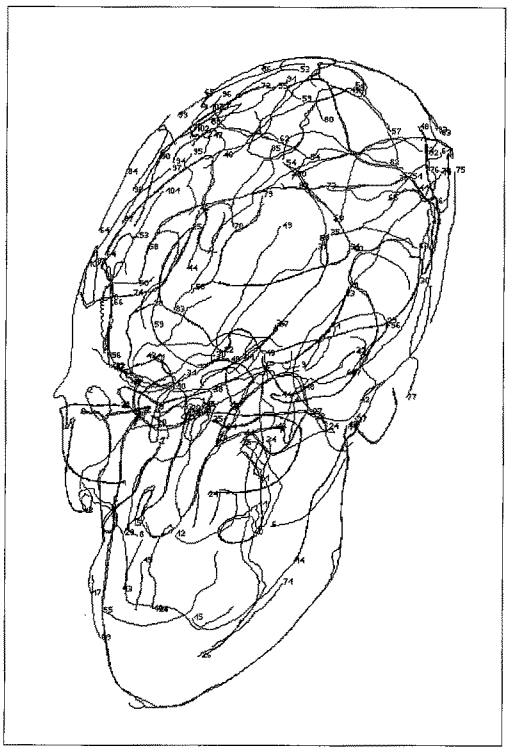


Fig. 16. (a) Isosurface on CT-scan data of arthur. (b) Crest lines on arthur (courtesy of Alexis Gourdon).



(a)

Fig. 17. (a) Crest lines on two 3-D images of arthur (courtesy of GE-CGR).



(b)

Fig. 17. (Continued). (b) Superposition of the lines of (a) with our software in 16.2 seconds on a DEC 5900: 42 lines out of 62 were identified. 262 points from different curves were matched out of 1618 points (view A) and 1516 points (view B).

5 Summary

We have presented a very efficient method based on geometric hashing that rigidly matches 3-D curves. We are still improving the computation of useful differential invariants on curves and surfaces.

We will next study the precision and the robustness of the matching procedure with data that has been calibrated. We will also investigate the possible generalization of the method to the case of small deformations.

6 Acknowledgments

We gratefully acknowledge the help of 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 and Alexis Gourdon for assistance. CT-scan images of skulls were provided by GE-CGR, Buc, France. Images of the vertebra were provided by J.L. Coatrieux and R. Collorec, Groupe Signaux et Images en Médicine, Université Rennes I.

Appendix A Penalization Parameters Optimization

We experienced two different methods:

1. *Generalized cross-validation* as in Craven and Wahba (1979), or Girard (1989). Knowledge of the uncertainty on the data is not necessary for this method.
2. *Newton method* already explored by Reinsch (1967, 1970) but presented here in a slightly different fashion. It uses explicitly the expected variance of the error between the data curve and its approximation.

A.1 Generalized Cross-validation

Suppose we remove one point x_k from the curve prior to do the spline approximation S, $S = B^t A_\tau^{-1} Bx$ (see section 2.3.3), so that we obtain a new approximation S_k. We expect the missing point x_k to be correctly extrapolated by our new approximation $S_k x_k$. The cross-validation function measures the extrapolation error that is done when each data point is successively removed:

$$v_0(\tau) = \sum_{k=0}^{n} \|S_k x_k - x_k\|^2$$

Of course, the cross-validation function V_0 depends on τ, as S depends on τ and our objective is to find the τ value that minimizes $V_0(\tau)$. A key fact is that matrixes $S_k x_k$ and Sx_k are related:

$$S_k x_k - x_k = \frac{Sx_k - s_k}{1 - S_{kk}}$$

This is because the approximation is obtained by solving the same system and by replacing the missing value x_k with the extrapolated value $S_k x_k$: $S_k x_k = Sx_k - S_{kk} x_k + S_{kk} S_k x_k$. Thus, the cross-validation function V_0 can be written as

$$V_0(\tau) = \sum_{k=0}^{n} \frac{\|Sx_k - x_k\|^2}{(1 - S_{kk})^2}$$

Now, because the computation of the values S_{kk} might be too costly, Craven and Wahba (1979) proposed to minimize the *generalized cross-validation* function $V(\tau)$, provided the diagonal term S_{kk} vary little with respect to k.

$$V(\tau) = \frac{\|Sx - x\|^2}{\left(1 - \frac{Tr(S)}{n}\right)^2}$$

Then, researchers like Silverman (1984) or Utreras proposed methods to approximate efficiently the trace of matrix S, $Tr(S)$. Further, a Monte Carlo method was introduced by Girard (1989) to estimate that trace.

We implemented that last Monte Carlo version of the generalized cross-validation technique. Thus, we applied the spline approximation to a random matrix w as well as to the data x for each τ value. We also experimented with the direct computation of S_{kk}, which was workable because our S matrixes were small enough (the number of control vertexes defining spline curves was between 10 and 50).

These two implementations produced very similar results: τ was optimized so as to best fit the spline to the data, with a very small approximation error, which corrupted the curvature and torsion estimations (see figure 18a). This behavior may be due to a high correlation of the error on our data, as we have recently noticed.

The next Newton method permits setting a higher approximation error σ (σ standard deviation value, as opposed to maximum error ϵ_∞) and to obtaining smoother curves with reliable curvature and torsion estimates.

A.2 Newton Method

We call $F_x(t)$ the residue $\|Sx - x\| = \|\Delta x\|$. In this method, the ideal value for τ is obtained when $F_x(\tau) = \sigma$, σ being given. Let us differentiate with respect to τ:

$$A_\tau = BB^t - \tau B'' B'^t$$

$$\frac{dA}{d\tau} = B'' B'^t$$

$$\frac{dA^{-1}}{d\tau} = -A^{-1} B'' B'^t A^{-1}$$

$$F_x(\tau) F_x'(\tau) = (Sx - x, -B^t A^{-1} B'' B'^t A^{-1} Bx)$$

$$= -(\Delta x, D^t Dx)$$

if we set $D = B'^t A^{-1} B$, Dx is the second derivative computed on the spline. Finally,

$$F_x(\tau) F_x'(\tau) = (D \Delta x, Dx)$$

thus

$$\frac{F_x(\tau) - \sigma}{F_x'(\tau)} = \frac{(\|\Delta x\| - \sigma)\|\Delta x\|}{(D \Delta x, Dx)}$$

This is Newton's correction amount. $F_x(\tau)$ is a monotonous increasing function since the quantity $(Dx, D \Delta x)$ is negative ($\|DSx\| \leq \|Dx\|$). Otherwise smoothing splines would not smooth at all!

If S is an exact projection (least squares without smoothing), we can write that $D \Delta x = DSx - Dx = Dx - Dx = 0$. Thus, with a suitable first guess for τ, and a σ value that approaches the real variance on the position of the data, the Newton method converges (see figure 18b). It may also diverge in a spectacular fashion.

Our method is different from Reinsch's (1967, 1970). Reinsch (1970) proves that the function $F_x(1/\tau)$ is concave inside the interval $]0, \infty]$ and notices that nothing can be told about $F_x(\tau)$, which we use. To see this, it suffices to differentiate $F_x(1/\tau)$ twice with respect to τ. Moreover, Reinsch explains that the optimization of $1/\tau$ is sometimes *too slow* to be useful and he proposes to improve it. We were satisfied with the results obtained with our method using $F_x(\tau)$, since it diverges only for inappropriate σ values.

We next consider the case of two smoothing parameters τ and ν. It might be possible to find the

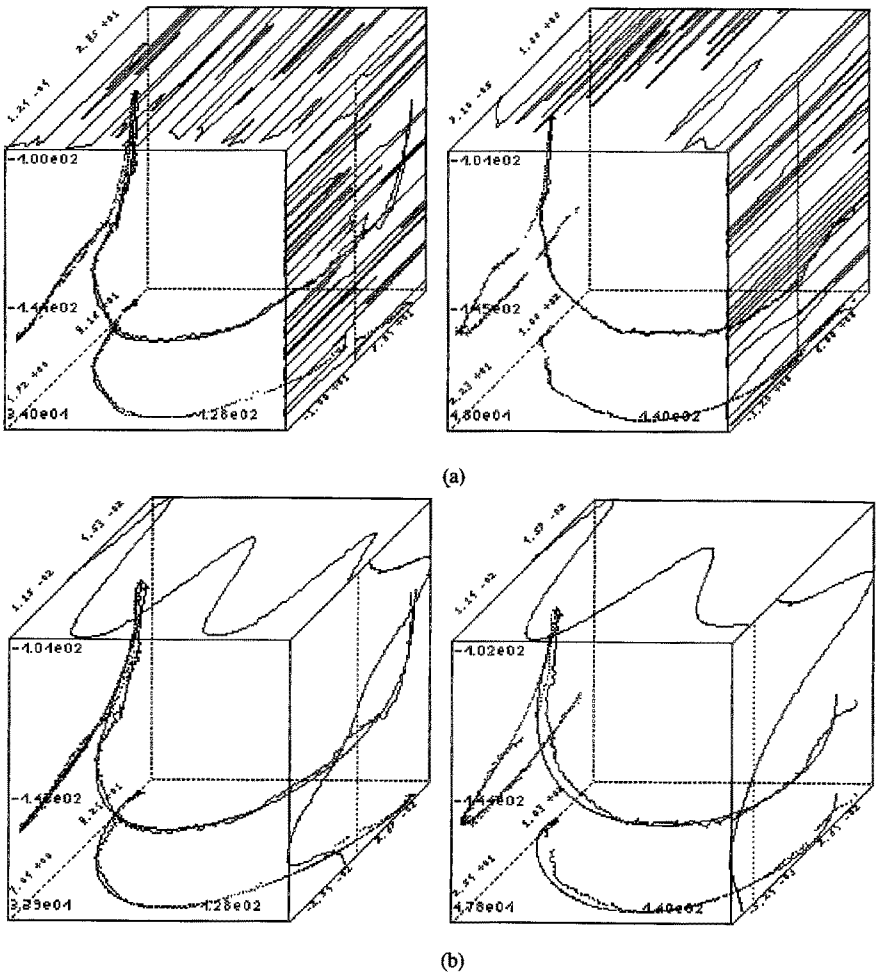


Fig. 18. (a) Approximations with the generalized cross-validation technique. The τ parameter is automatically set so that the curve lies close to the data. On the other hand, the oscillations necessary for this fit corrupt the curvature and torsion estimations, on top and right panels of the plots. (b) With Newton's method, the τ obtained with the knowledge on the variance *sigma* of the data provide in smooth approximations. Thus we obtain reliable curvature and torsion values, that can be matched from the two views A and B. Notice that we use a third-order derivative penalty that permits us to regularize both curvature and torsion. (*Note:* All plots are the same scale.)

minimum of a two-variables function $F_x(\tau, \nu)$ with the knowledge of the gradient (F_τ, F_ν). We note the operator $\Pi = N_3^t B_3'' A_3^{-1} B_3$, Πx is the $n + 1$-vector of scalar products tangent-$\mathbf{n_i}$ and we have

$$F(\tau, \nu) F_\nu = (\Pi \Delta x_3, \Pi x_3)$$

We could also set one of the two variables τ and ν and optimize the second one with the previous equation of the residue.

However, we did all our experiments with a unique smoothing parameter. Figure 18 shows the results of our algorithms on two significant corresponding curves from views A and B of our skull. With the cross-validation technique (see figure 18a), we obtain a good fit and a poor curvature estimation. With the Newton descent, associated with an adequate residue, we obtain a smoother approximation together with useful curvature and torsion values (see figure 18b).

Appendix B A Bayesian Analysis for Model Indexing

As in (Rigoutsos & Hummel 1991) we consider the probability $P(c\ \tau)$ of finding an entry in hash table at the location (c, τ), as well as the conditional probability $P(c, \tau/M_i)$, assuming that the model M_i appears in the scene.

With Bayes' rule, we can write

$$\prod P(M_i/c, \tau) = \prod \frac{P(c, \tau/M_i)}{P(c, \tau)} P(M_i)$$

(Π is the product over all (c, τ) values involved in model M_i.)

The log-probability $\log [P(c, \tau/M_i)/P(c, \tau)] > 0$ measures the increase of the likelihood of the hypothesis that we recognize model M_i from the scene, given the evidence that we observe (c, τ) values. In hash-space, as visualized in figure 19, we expect to observe a background distribution of measures (c, τ) as well as peaks which attest that model M_i is present in the scene.

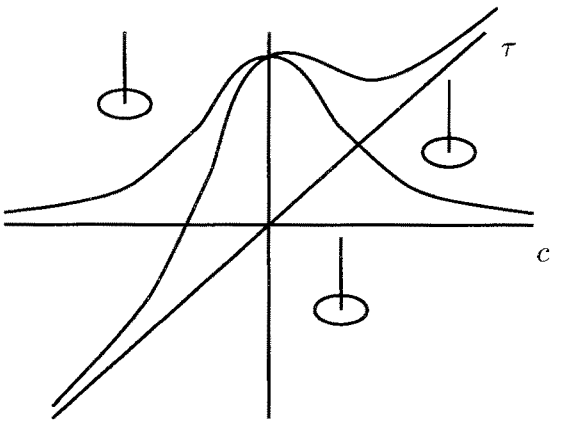


Fig. 19. The background distribution of (c, τ) with peaks corresponding to models present in the scene (inspired by Rigoutsos & Hummel 1991).

Once all values (c, τ) observed along data curves are processed, as described in section 3.1.3, the consistent matchings of pairs (model point, scene point) will score the highest likelihood values:

$$\sum \log \frac{P(c, \tau/M_i)}{P(c, \tau)}$$

(Σ is the sum over all (c, τ) values that contribute to the match.)

The likelihood is bounded above because there is a limited number of points that can be matched between the models and the scene. However, if we have new evidence (new points), the method might select a new peak in transformation space that supersedes the previous one. This is exactly what happens in figure 12e: With the incorporation of the orbital crest line, we evolve from a local match to a global match. No dynamics are involved in this process: we simply pick a transformation that scores a higher vote. Alternatively, too much evidence will not necessarily improve the transformation. In figure 12f, all other curves have been incorporated, but the transformation is pretty much the same as in figure 12e.

Thus, local minima of potential as when "one is stuck in a local match" are not a big issue in this matching technique. Moreover, we could set a desirable likelihood threshold and infer the minimum number of points that is needed to score this value. We could even try to link this likelihood of match with the precision on the transformation that is estimated with the Extended Kalman Filter (see Section 3.2.3).

Acknowledgment

This work was supported in part by a grant from Digital Equipment Corporation.

Note

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 Ayache 1991).

References

Arbogast, E. 1990. La représentation des contours et leur segmentation. Tech. Rept. RR 115, LIFIA, juillet.

Ayache, N. 1991. *Artificial Vision for Mobile Robots—Stereo-Vision and Multisensory Perception*. MIT Press: Cambridge, MA.

Ayache, N., and Faugeras, O.D. 1986. Hyper: A new approach for the recognition and positioning of two-dimensional objects, *IEEE Trans. Patt. Anal. Mach. Intell.* 8(1):44–54.

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

Ayache, N., Boissonnat, J.D., Cohen, L., Geiger, B., Levy-Vehel, J., Monga, O., and Sander, P. 1990. Steps toward the automatic interpretation of 3-D images. In H. Fuchs, K. Hohne, and S. Pizer, eds., *3D Imaging in Medicine*, NATO ASI Series, Springer-Verlag: New York, pp. 107–120.

Bartels, R., Beatty, J., and Barsky, B., 1987. *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. Morgan Kaufmann: San Mateo, CA.

Benayoun, S., Monga, O., Guéziec, A., and Ayache, N. 1992. Using surface curvatures for 3-D image registration, *Proc. 11th Conf. Comput. Appl. Radiol.*, Baltimore, MD.

Besl, P. and McKay, N. 1992. A method for registration of 3-D shapes, *IEEE Trans. Patt. Anal. Mach. Intell.*, 14(2):239–256.

Bohm, W., Farin, G., and Kahmann, J. 1984. *A Survey of Curve and Surface Methods in CAGD*, Elsevier Science Publishers: North Holland, in Computer Aided Geometric Design, pp. 1–60.

Cinquin, P. 1981. Splines unidimsionnelles sous tension et bidimensionnelles paramétrées, Thèse de Doctorat de troisième cycle, Université de St-Etienne, Octobre.

Cinquin, P. 1987. Application des fonctions-spline au traitement d'images numériques. Thèse de Doctorat de mathématiques, Septembre.

Cohen, I. Cohen, L.D., and Ayache, N. 1991. Introducing deformable surfaces to segment 3-D images and infer differential structures, *Proc. Conf. Comput. Vis. Patt. Recog.*, Hawaii, June.

Cohen, L.D. and Cohen I. 1990. A finite element method applied to new active contour models and 3-D reconstruction from cross sections, *Proc. 3rd Intern. Conf. Comput. Vis.*, Osaka, Japan, December.

Craven, P. and Wahba, G. 1979. Smoothing noisy data with spline functions. *Numerische Mathematik* 31:377–403.

Cutting, C.B. 1989. Applications of computer graphics to the evaluation and treatment of major craniofacial malformation. In Udupa and Herman, eds., *3-D Imaging in Medicine*, CRC Press: Boca Raton, FL.

de Boor, C. 1978. *A practical Guide to Splines*. Springer-Verlag: New York.

de Hoog, F.R. and Hutchinson, M.F. 1987. An efficient method for calculating smoothing splines using orthogonal transformations, *Numerische Mathematik* 50:311–319.

do Carmo, M.P. 1976. *Differential Geometry of Curves and Surfaces*. Prentice-Hall: Englewood Cliffs, NJ.

Duda, R., and Hart, P. 1973. *Pattern Classification and Scene Analysis*. Wiley: New York.

Farin, C. 1988. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press: San Diego, CA.

Girard, D. 1989. A fast "monte-carlo cross-validation procedure" for large least squares problems with noisy data, *Numerische Mathematik* 56:1–23.

Grimson, W.E.L., and Huttenlocher, D.P. 1991. On the verification of hypothesized matches in model-based recognition, *IEEE Trans. Patt. Anal. Mach. Intell.* 13(12):1201–1213, December.

Grimson, W.E.L. and Lozano-Perèz, T. 1984. Model-based recognition and localization from sparse range or tactile data, *Intern. J. Robot. Res.* 3(3):3–35.

Grossman, M. 1971. Parametric curve fitting, *Computer Journal* 14:169–172.

Guéziec, A. 1993. Large deformable splines, crest lines and matching, *Proc. 4th Intern. Conf. Comput. Vis.*, Berlin, May.

Guéziec, A., and Ayache, N. 1992. Smoothing and matching of 3-D-space curves, *Proc. 2nd Europ. Conf. Comput. Vis.*, Santa Margherita Ligure, Italy, May.

Herlin, I., and Ayache, N. 1992. Features extraction and analysis methods for sequences of ultrasound images, *Proc. 2nd Europ. Conf. Comput. Vis.*, Santa Margherita Ligure, Italy, May.

Holladay, J.C. 1957. Smoothest curve approximation, *Math. Tables Aids Computation* 11:233–243.

Kishon, E., Hastie, T., and Wolfson, H. 1990. 3-D curve matching using splines, *Proc. Ist Europ. Conf. Comput. Vis.*, Antibes, France, April.

Lancaster, P., and Salkauskas, K. 1986. *Curve and Surface Fitting, an Introduction*. Academic Press, San Diego, CA.

Malandain, G., Bertrand, G., and Ayache, N. 1991. Topological segmentation of discrete surface structures, *Proc. Conf. Comput. Vis. Patt. Recog.*, Hawaii, June.

Marin, S.P. 1984. An approach to data parametrization in parametric cubic spine interpolation, *J. Approx. Theory* 41:64–86.

Monga, O., Ayache, N., and Sander P. 1991. From voxels to curvature, *Proc. Conf. Comput. Vis. Patt. Recog.*, Hawaii, June.

Monga, O., Benayoun, S. and Faugeras, O.D. 1992. Using third order derivatives to extract ridge lines in 3-D images, *Proc. Conf. Comput. Vis. Patt. Recog.*, Urbana Champaign, June.

Monga, O., Deriche, R. and Rocchisani, J.-M. 1991. 3-D edge detection using recursive filtering: Application to scanner images, *Comput. Vis. Graph. Image Process.*, January.

Nilson, E.N., Ahlberg, J.H., and Walsh, J.L. 1967. *The Theory of Splines and Their Applications*. Academic Press: New York.

Pavlidis, T. 1977. *Structural Pattern Recognition*. Springer-Verlag: New York.

Plass, M. and Stone, M. 1983. Curve fitting with piecewise parametric cubics, *Siggraph*, pp. 229–239, July.

Reinsch, C.H. 1967. Smoothing by spline functions, *Numerische Mathematik* 10:177–183.

Reinsch, C.H. 1970. Smoothing by spline functions II, *Numerische Mathematik* 16:451–454.

Rigoutsos, I. 1992. *Massively parallel Bayesian object recognition*. Ph.D. thesis, New York University.

Rigoutsos, I., and Hummel, R. 1991. Robust similarity invariant matching in the presence of noise: A data parallel approach, *Proc. 8th Israeli Conf. Artif. Intell. Comput. Vis.*, December.

Saint-Marc, P. and Medioni, G. 1990. B-spline contour representation and symmetry detection, *Proc. Ist Europ. Conf. Comput. Vis.*, Antibes, April.

Schwartz, J.T., and Sharir, M. 1987. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves, *Intern. J. Robot. Res.* 6:229–244.

Silverman, B.W. 1984. A fast and efficient cross-validation method for smoothing parameter choice in spline regression, *J. Amer. Statist. Assoc.* 79:584–589, September.

Stein, F. 1991. Structural hashing: Efficient 3-D object recognition, *Proc. Intern. Conf. Comput. Vis. Patt. Recog.*, Hawaii, June.

Thirion, J.P. and Gourdon, A. 1992. The 3-D matching lines algorithm and its application to crest lines extraction, Tech. Rept. 1672, INRIA, May.

Thirion, J.P. and Gjoourdon, A. 1993. The matching lines algorithm: new results and proofs, Tech. Rept. 1881, INRIA.

Thirion, J.P., Gourdon, A., Monga, O., Guéziec, A., and Ayache, N. 1992. Automatic registraction of 3-D CAT-SCAN images using surface curvature, *Proc. 14th Annu. Intern. Conf. IEEE Engineer. Med. Biol. Soc. Satellite Symposium on 3-D Advanced Image Processing in Medicine*, Rennes, France. November.

Thompson, D.W. and Mundy, J.L. 1987. 3-D model matching from an unconstrained viewpoint, *Proc. Intern. Conf. Robot. Autom.*, pp. 208–220.

Wang, Shin, Ywan and Ferrari, L.A. 1990. Automatic data visualization using spline functions. Unpublished manuscript, Department of Electrical and Computer Engineering, University of California, Irvine CA 92717.

Zhang, Zhengyou. 1992. Iterative point matching for registration of free-form curves, Tech. Rept. 1658, INRIA, March.