Smoothing and Matching of 3-D Space Curves *

André Guéziec and Nicholas Ayache

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

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

Our method builds upon the seminal work of [9], 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 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.

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 [1]. Characteristic curves describe either topological singularities such as surface borders, hole borders, and simple or multiple junctions, etc., (see [10]), or differential structures, such as ridges, parabolic lines, and umbilic points [11].

* This work was financed in part by a grant from Digital Equipement Corporation. General Electric-CGR partially supported the research that provided ridge extraction software.

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 [12]. These curves can be used to serve as a reference identifying positions and features of the skull and to establish landmarks to match skulls between different individuals, yielding a standard approach for complex skull modeling [4].

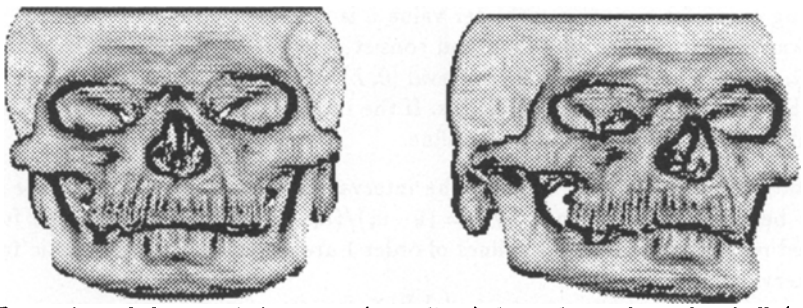

Fig. 1. Extraction of characteristic curves (crest lines) from the surface of a skull (using two different X-ray Scanner images)

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 intrinsic differential features of second and third order (curvature and torsion). Given two curves, we then wish to find, through a matching algorithm, the longest common portion, up to a rigid transformation. From three possible approaches, specifically: prediction-verification, accumulation and geometric hashing, we retained the third one whose complexity is sublinear in the number of models. We call it an indexation method, and introduce logical extensions of the work of [9, 15, 3]. Our work is also closely related to the work of [2, 6, 16] on the identification and positionning of 3D objects.

In Section 2, we discuss approaches to fitting curves to collections of voxels (points) in 3D imagery. In Section 3, we implement a matching system based on the indexation (geometric hashing), whose complexity is sublinear in 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). We finally introduce alternative invariants for hashing. In sum, we considerably extend previous indexation-based curve-matching methods. In Section 4, we provide experimental results obtained using real data.

2 Approximation of Noisy Curves with B-Splines

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 [11]).

B-spline curves, which include the class of polygonal curves, 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 [8]. There is an

extensive literature on B-splines We provide a very brief introduction, using the notation of [3, 15]. Given a sequence of $n+1$ points $P_i(x_i, y_i, z_i)$, $i = 0..n$ in 3-space, a $\mathcal{C}^{K-2}$ approximating B-spline consists of the following components:

1. A control polygon of $m+1$ points is given, such that $V_j(X_j, Y_j, Z_j)$,$j = 0..m$ 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 $\mathcal{C}^{K-2}$. The location in 3-space of the approximating curve for a given parameter value $\bar{u}$ is given by: $Q(\bar{u}) = \sum_{j=0}^{m} V_j B_{j,K}(\bar{u})$.
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 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 order 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 $\mathcal{C}^1$, cubic splines $(B_{j,4})$, are $\mathcal{C}^2$, etc. 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.

2.1 A Previous Approximation Scheme

We next recall a classic approximation scheme due to Barsky [3]. This scheme has been used by St-Marc and Médioni [15] for curve matching. Our emphasis is on the shortcomings of the approach for our objectives and on proposed modifications.

Given $n+1$ data points $P_i(x_i, y_i, z_i)$, $i = 0..n$, we seek $m+1$ control vertices 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 $Q(\bar{u})$ 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 closes to P_i. Thus, the criterion to minimize is:

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

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. Thus as an estimate, we can use $\tilde{u}_i = \sum_{k=0}^{i-1} ||P_{k+1} - P_k||$. If B is the $m+1$ by $n+1$ matrix of the $B_{j,K}(\tilde{u}_i)$, X the $m+1$ by 3 control vertices matrix and x the $n+1$ by 3 matrix of data points coordinates, Δ_1 can we written as $||B^t X - x||^2$. Differentiating with respect to X leads to:

$$BB^t X - Bx = 0 \text{ or } AX = Bx.$$

Because $X^t BB^t X = ||B^t X||^2$, we know that A and B have the same rank. Thus if $m \leq n$ A is positive definite up to numerical error. If the approximating curve is not a closed curve then A is a band matrix of band size K and X can be determined in linear time with respect to m with a Choleski decomposition.

In working with this method, we have observed that $m+1$, 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. 2a), we reconstruct Fig. 2b 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. 2c and 2d are obtained.

2.2 Improvements

Better Knot Distribution. The vertices of an approximating polygonal path will concentrate around locations of high curvature [13]. 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 having a well defined number of knots $m + K$ (and consequently of vertices $m + 1$), will more closely approximate these portions of the curve. In order to cope with noise, the tolerance level of the polygonal fit must exceed the standard deviation on the position of the points.

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 P_i onto the approximating polygonal path and consider the relative position of the projected points in terms of total chordlength of the path.

Improved Distance Estimates [14]. 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:

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

We update $\tilde{u}_i$ by a Newton Raphson iteration, using the quantity $\delta_i = F_i(\tilde{u}_i)/F_i'(\tilde{u}_i)$. For a detailed calculation, the reader may refer to [7]. Despite their apparent complexity, these computations are not very expensive, since $B'_{j,K}$ and $B''_{j,K}$ were necessarily calculated before $B_{j,K}$ (by the recursive definition). Moreover, once all $\tilde{u}_i$ are updated by the amounts δ_i, we must once again solve the linear system for new control vertices $\{V_j\}$.

Minimization of Curvature. Cubic B-splines minimize, among all interpolants, the norm of the second derivative[8]. 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:

$$\Delta_2 = \frac{\Delta_1}{{\sigma_1}^2} + \frac{\sum_{i=0}^{i=n} ||Q''(\tilde{u}_i)||^2}{{\sigma_2}^2}, \text{ where } \begin{cases} {\sigma_1}^2 = \text{Var}(||Q(\tilde{u}_i) - P_i||), \\ {\sigma_2}^2 = \text{Var}(||Q''(\tilde{u}_i)||), \end{cases}$$

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. Since the second derivative values are linear in terms of control vertices, Δ_2 is again quadratic, the construction and complexity are as before, and the result is a spline. Fig. 2d illustrates results of minimizing Δ_2.

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:

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

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 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. In [11], a study on the stability of such measurements is provided. 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 regularization parameters $\tau = (\sigma_2/\sigma_1)^2$ and $\nu = (\sigma_3/\sigma_1)^2$ are arbitrarily chosen so that Δ_2 and Δ_3 have no unit. We will describe and compare in a forthcoming report automatic methods to optimize τ and ν.

3 Indexing for Curve Model Matching

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.

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 the models curves have been sampled according to the original sampling in the image.

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 $D_{i,j,l}$ (see [7]) 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 $(M_i, D_{i,j,l})$. This is NOT Hough transform. In Hough

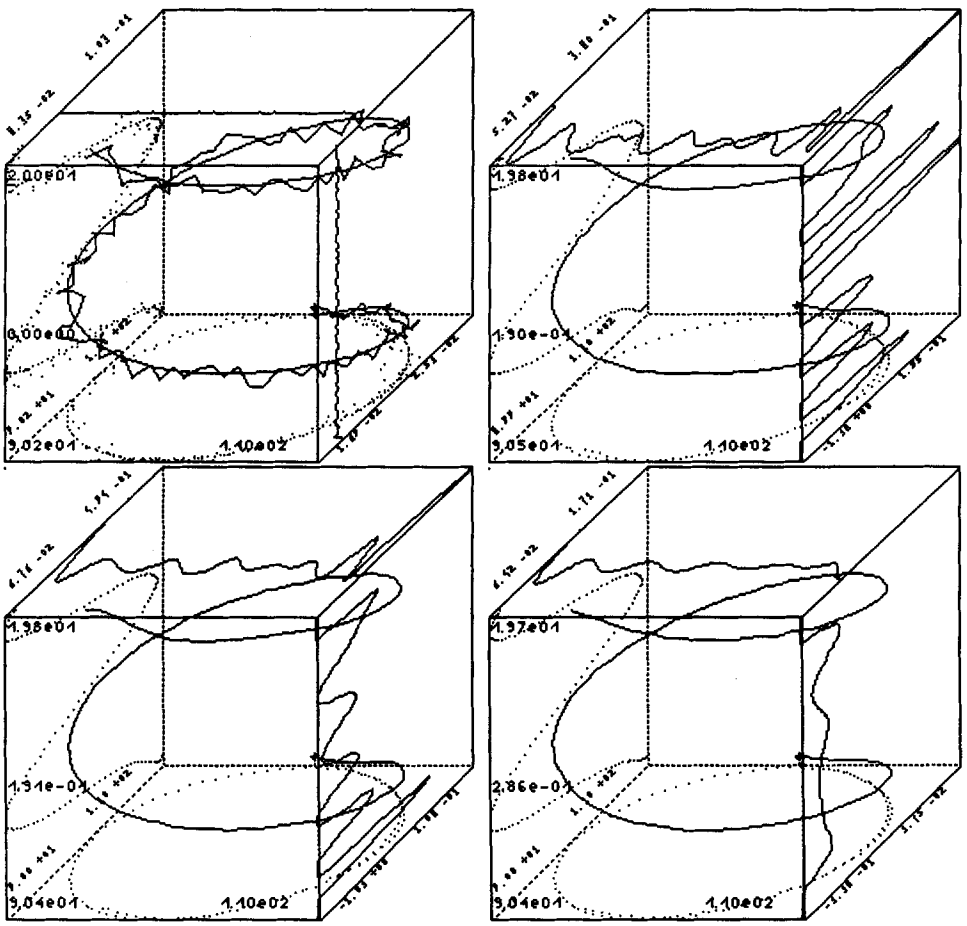

Fig. 2. a. Top left: Noise is added to a helix, and points are sampled with a limitation on the distance between successive points. The curvature and torsion are plotted in the top and the right panels of the cube, as a function of arclength. In a perfect reconstruction, the curvature and torsion would be constant.
b. Top right: 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. Bottom left: A more precise estimate of model-data distances improves the estimation of curvature and torsion.
d. Bottom right: The constraint on the second derivative also improves the estimation.

transform, a hypothesis votes for a hyperplane in parameter space. Thus cluster detection is inefficient (see [17]). We instead vote for a single point.

After processing all of the n 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, 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.

This description of the method of indexation is essentially the "geometric hashing" method of Kishon and Wolfson [9], updated in one important aspect. They use a polyg-

onal 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. 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 [7].

3.1 Enhancements to the Indexation Method

Indexation Table Quantization. Guided by [5], we collect statistics based on experiments with simulation and real data, described in [7]. 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.

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.

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.

Alternative Geometric Invariants for Matching. Suppose that we are given a reference point B on a model curve, and consider the points P on the same curve. 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, 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, 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. We investigate in [7] the utility of these various invariants, and observe that θ_t and $\mathbf{u}_t$ are more stable than torsion, and have a greater discrimination power than $||\mathbf{u}||$.

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 extremal curvature basis points* B along the curve. In this way, the information about the model curves are stored into a three-dimensional table, indexed by $(c, \theta_t, \mathrm{u}_t)$. 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.

For the recognition algorithm, a 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 $(c, \theta_t, \mathrm{u}_t)$ 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 to **scattered points** associated with curvature information and a local reference frame. Experimental results are reported in the next section.

4 Results

Using two views (A and B) of a skull from real X-ray Scanner data, we used existing software (see [12]) to find ridge points, and then fed these points into the curve smoothing algorithm of Section 2. For each view, the sub-mandibular rim, the sub-orbital ridges, the nose contour and other curves were identified.

Using the indexation algorithm of Section 3.1, we preprocessed all curves from A, also in the reverse orientation if necessary (to cope with orientation problems), and built the indexation table, based on measurements of $(c, \theta_t, \mathrm{u}_t)$ along the curves. Applying the indexation-based recognition algorithm, curves from B were successfully identified. The resulting transformations were applied to all curves from B and superimposed matches (model, scene) appear in Figs. 3b to 3d. We next run our algorithm on the chin and right orbit curves considered as one single curve (Fig. 3e) and finally on all curves simultaneously (Fig. 3f). CPU times on a DEC-workstation (in seconds) for recognition and positionning are summarized in the following table. It confirms the linear time hypothesis.

scene curve	nose contour	right orbit	left orbit	chin	chin - orbit	all curves from B
CPU time	1.085	0.964	1.183	2.577	3.562	9.515

Note that incorporating more curves increases the likelihood of the match. We thus start from a local curve match and end up with one global rigid transformation.

We then experimented the matching using scattered points (several hundreds) on the surface of the objet, selected for the high curvature value on the surface and associated with a surface frame [12](Fig 3g). Last, we registered the entire skull by just applying the transformation that superimposed the two submandibular curves. Incorporating the match of the orbital ridge curves, we improved the overal rigid transformation estimate, resulting in a more precise correspondence (Fig 4).

References

1. 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.
2. N. Ayache and O.D. Faugeras. Hyper: A new approach for the recognition and positioning of two-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):44–54, January 1986.

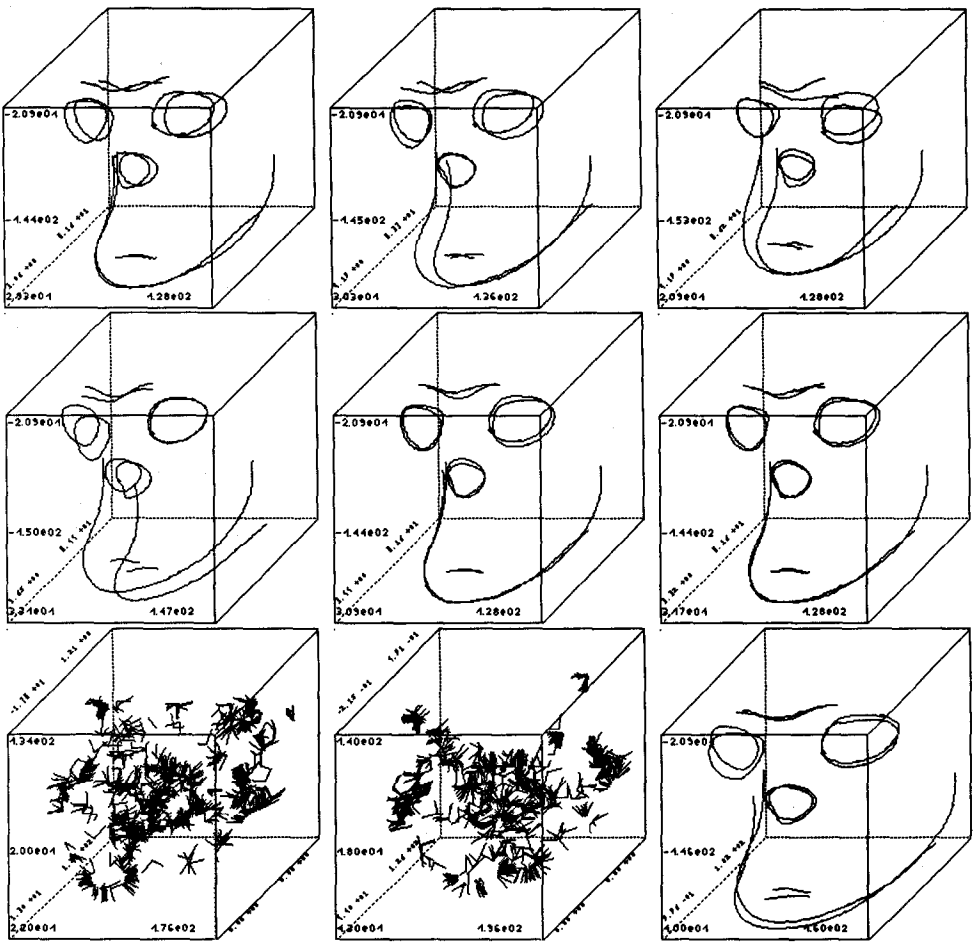

Fig. 3. a. Top left: The successful matching of the two sub-mandibular curves, superimposed. (Note that the occlusion and translation of the second view are handled automatically).
b. Top middle: Nose contours matched. **c. Top right:** Right orbits matched.
d. Middle left: Left orbits matched. **e. Middle:** Chin-orbit matched simulaneously.
f. Middle right: All curves matched simulaneously.
g. Bottom: The matching algorithm is successfully applied (bottom right) to scattered points associated to A (bottom left) and B (bottom middle), 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 maps).

3. R. Bartels, J. Beatty, and B. Barsky. *An introduction to splines for use in computer graphics and geometric modeleling.* Morgan Kaufmann publishers, 1987.
4. Court B. Cutting. Applications of computer graphics to the evaluation and treatment of major craniofacial malformation. In Jayaram K.Udupa and Gabor T. Herman, editors, *3D Imaging in Medicine.* CRC Press, 1989.
5. W. Eric L. Grimson and Daniel P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1201–1213, December 1991.
6. W.E.L Grimson and T. Lozano-Perèz. Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35, 1984.

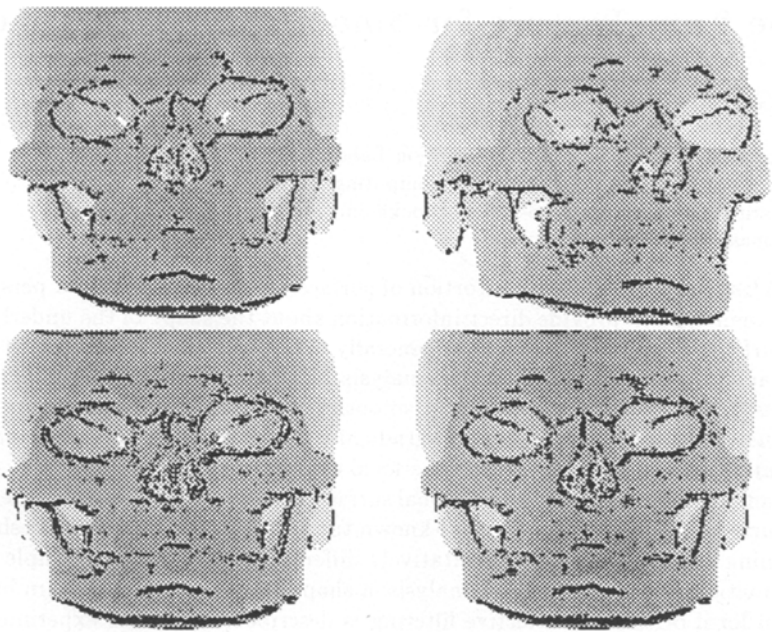

Fig. 4. Registrating the ridges : top row shows the ridges extracted of a skull scanned in position *A* (top left) and position *B* (top right). Figure in the bottom left shows the superposition of the ridge points, obtained after transforming the points of the second view according to the transformation discovered by matching the sub-mandibular curves. The best correspondences are along the chin points. Figure in the bottom right shows the improved transformation obtained with the addition of left sub-orbital curves.

7. A. Guéziec and N. Ayache. Smoothing and matching of 3d-space curves. Technical Report 1544, Inria, 1991.
8. J.C. Holladay. Smoothest curve approximation. *Math. Tables Aids Computation*, 11:233–243, 1957.
9. E. Kishon, T. Hastie, and H. Wolfson. 3-d curve matching using splines. Technical report, AT&T, November 1989.
10. G. Malandain, G. Bertrand, and Nicholas Ayache. Topological segmentation of discrete surface structures. In *Proc. International Conference on Computer Vision and Pattern Recognition*, Hawai,USA, June 1991.
11. O. Monga, N. Ayache, and P. Sander. From voxels to curvature. In *Proc. International Conference on Computer Vision and Pattern Recognition*, Hawai,USA, June 1991.
12. Olivier Monga, Serge Benayoun, and Olivier D. Faugeras. Using third order derivatives to extract ridge lines in 3d images. In *submitted to IEEE Conference on Vision and Pattern Recognition*, Urbana Champaign, June 1992.
13. T. Pavlidis. *Structural Pattern Recognition*. Springer-Verlag, 1977.
14. M. Plass and M. Stone. Curve fitting with piecewise parametric cubics. In *Siggraph*, pages 229–239, July 1983.
15. P. Saint-Marc and G. Medioni. B-spline contour representation and symmetry detection. In *First European Conference on Computer Vision (ECCV)*, Antibes, April 1990.
16. F. Stein. Structural hashing: Efficient 3-d object recognition. In *Proc. International Conference on Computer Vision and Pattern Recognition*, Hawai,USA, June 1991.
17. D. W. Thompson and J. L. Mundy. 3-d model matching from an unconstrained viewpoint. In *Proc. International Conference on Robotics and Automation*, pages 208–220, 1987.