3D-2D projective registration of free-form curves and surfaces

Jacques Feldmar, Nicholas Ayache, Fabienne Betting

3D–2D Projective Registration of Free-Form Curves and Surfaces

JACQUES FELDMAR,* NICHOLAS AYACHE, AND FABIENNE BETTING

INRIA Sophia, Projet EPIDAURE, 2004 route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex, France

Some medical interventions require knowing the correspondence between an MRI/CT image and the actual position of the patient. Examples occur in neurosurgery and radiotherapy, but also in video surgery (laparoscopy). We present in this paper three new techniques for performing this task without artificial markers. To do this, we find the 3D–2D projective transformation (composition of a rigid displacement and a perspective projection) which maps a 3D object onto a 2D image of this object. Depending on the object model (curve or surface), and on the 2D image acquisition system (X-Ray, video), the techniques are different but the framework is common:

• We first find an estimate of the transformation using bitangent lines or bitangent planes. These are first order semi-differential invariants.
• Then, introducing the normal or tangent, we define a distance between the 3D object and the 2D image, and we minimize it using extensions of the Iterative Closest Point algorithm.
• We deal with the critical problem of outliers by computing Mahalanobis distances and performing generalized χ^2 tests.

Results are presented on a variety of real medical data to demonstrate the validity of our approach.   © 1997 Academic Press

1. INTRODUCTION

Medical images are commonly used to help establish a correct diagnosis. As they contain spatial information, both anatomical and functional, they can also be used to plan therapy and even in some cases to control the therapy. A recent overview of such research can be found in [2] and in [32], a spectacular use of planning and control of therapy using medical images and robots can be found in [7, 21, 31] for surgery and [28] for radiotherapy. Some basic techniques involved in these problems are presented in [16].

One of the most difficult tasks is to register, if possible in real time, a video or X-ray image of the patient (intraoperative image) with a preoperative image (MRI or CT). Basically, one must find the 3D–2D projective transformation (composition of a rigid displacement and a perspective projection) which maps a 3D object onto a 2D image of this object, the relative positions of the 3D object and the 2D sensor being unknown.

This problem can be solved with artificial markers visible in both images,[1] but this produces unacceptable constraints (e.g., preoperative images must be taken on the same day as the operation and stereotactic frames are very painful and can prevent a free access by the surgeon).

Recently, computer vision techniques have been proposed to solve this registration problem without artificial markers. Grimson [17] uses an intermediate laser range finder, which provides a 3D description of the patient's surface. This surface is then matched against the surface of the segmented corresponding surface in the volumetric medical image. As the laser range finder is calibrated with respect to the camera, the medical image can be fused with the video image. Colchester [9] and Kanade [29] also developed *reconstruction/rigid registration* frameworks. The approach adopted by Schweikard [28] is a 2D correlation scheme between the radiography and precomputed simulated radiographies of a MRI image. Another correlation technique is presented in [22] and a mutual information maximization approach is proposed in [35]. Finally, Lavallee [21] uses the occluding contours of the 3D object in the 2D image to perform the 3D–2D registration task.

This paper is a contribution to this new and exciting research field. We introduce three different techniques for 3D–2D registration, depending on the particular clinical problem. We first present in Section 2 an approach which makes use of passive stereo to reconstruct the surface. The problem is in this case *3D–3D surface rigid registration.* Then, in Section 3 we present a framework for finding the geometric transformation between a 3D image of vessels and a 2D radiography of those vessels. This is the *3D–2D curve registration problem.* Finally, in Section 4, we present a technique which allows us to find the transformation between a 3D surface and an image of the surface, by using the silhouette. This is the *3D–2D surface registration problem.*

---

* E-mail: Jacques.Feldmar@sophia.inria.fr.

[1] The problem is then very similar to the camera calibration problem.

For the three registration problems addressed in this paper, we have developed a *unified framework*. The subsections correspond to each step of this approach:

• First, we find an *initial estimate* of the geometric transformation. To do that, we use bitangent planes or bitangent lines, which are semidifferential invariants[2] [15] involving only first-order derivatives. Hence, our framework does not depend on the initial relative positions of the objects.

• Then, we find an *accurate transformation* by defining a distance between the 3D and the 2D object (which is a compromise between the spatial distance and the difference in normal or tangent orientation) and by minimizing it using two-step minimization algorithms which are extensions of the Iterative Closest Point algorithm presented in [4, 36]. The use of extended Kalman filters allows us to take into account the critical problem of outliers, computing Mahalanobis distances, and performing generalized χ^2 tests.

2. 3D–3D SURFACE RIGID REGISTRATION

The approach presented in this section, to find the correspondence between a 3D image and the actual position of the patient, is related to [9, 17, and 29]. We also split the problem into two stages: reconstruction and rigid registration. But the way we perform the surface reconstruction is different. We use a passive stereo system developed by Devernay [11] within the Robotvis group at INRIA. The result is a dense description using points and normals of the patient's surface. The coordinates of these points and normals are expressed in the camera frame. Because the transformation which maps the reconstructed image to the camera image is known, the problem is to find the transformation between the MRI/CT image and the reconstructed surface.

In order to find this transformation, we extract from the MRI/CT image the surface of the patient's face by combining the techniques described in [24] and in [33]. Hence, we also get a description by points and normals, and the rigid registration problem[3] is the following:

> Given two surfaces described by points and normals, find the rigid displacement that best superposes these two surfaces.

As pointed out in [17], this algorithm must not depend
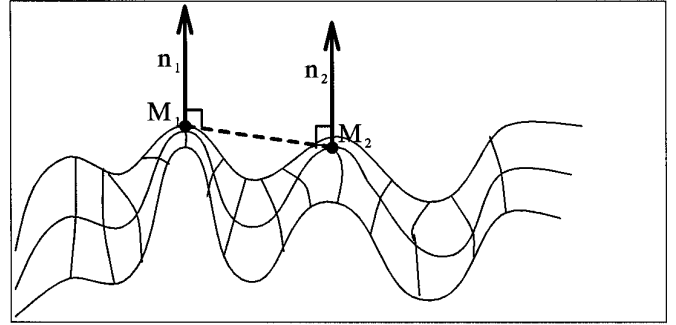

FIG. 1. A surface and two bitangent points M_1 and M_2. Let $\mathbf{n}_1$ and $\mathbf{n}_2$ be the normals at these points. M_1 and M_2 are bitangent if the plane defined by $(M_1, \mathbf{n}_1)$ and the plane defined by $(M_2, \mathbf{n}_2)$ are the same. Another definition is that $\mathbf{n}_1$ and $\mathbf{n}_2$ are identical and that the line $M_1 M_2$ is orthogonal to these two vectors.

on the initial relative positions of the surfaces, it must be accurate and deal with outliers.

2.1. *Finding an Initial Estimate of the Rigid Displacement*

The basic idea is to compute independently on each surface the set of pairs of points sharing the same tangent plane (see Fig. 1). We call such pairs *bitangent points*. As mentioned in Section 1, they correspond to semi-differential invariants [15]. The technique for computing these pairs is described in Appendix A.1. We simply note here that the algorithm is quasi-linear in the number of points describing the surface, and, because it involves only derivatives of order 1, the bitangent points calculation is quite stable.

In the ideal case, because the distance between the two bitangent points is invariant under rigid displacement, the following algorithm would be very efficient to rigidly superpose a surface S_1 on a surface S_2:

(1) Choose a pair P_1 of bitangent points on S_1. Let $d(P_1)$ be the distance between the two points.

(2) Compute the set $SameDistance(P_1)$ of pairs of bitangent points on S_2 such that the distance between the two bitangent points is equal to $d(P_1)$.

(3) For each pair P_2 in $SameDistance(P_1)$, compute the two possible rigid displacements corresponding to the superposition of the two pairs P_1 and P_2 and of their normals. Stop when the rigid displacement which superposes S_1 on S_2 is found.

In practice, corresponding pairs of bitangent points cannot be exactly superposed because of the point discretization error and because of the error in the computation of the normal. Moreover, only a part of the reconstructed surface S_1 may be superposed on the patient's surface ex-

---

[2] The basic idea of semidifferential invariants is to use several points to define invariant quantities. The advantage of this approach is that it involves lower order derivatives lower than the differential invariants computed at a single point, which make them more stable. The disadvantage is that the complexity of the algorithms often highly increases with the number of points used to define the semidifferential invariants. Hence, the choice of the invariants must be a compromise between the order of the involved derivatives and the complexity of the underlying algorithm.

[3] Note that this problem has many other applications (see [34]).

tracted from the MRI image S_2. So, the actual algorithm is slightly more complex, but is basically as stated. The details can be found in Appendix B.1, and an analysis of the complexity of the search for an initial estimate can be found in Appendix B.2. We simply note that the complexity of the algorithm is quasi-linear in the number of points on the surfaces and that the risk of stopping the algorithm with a wrong initial estimate decreases extremely quickly with the number of points (when the two surfaces actually show some overlapping regions up to a rigid displacement).

2.2. *The Iterative Closest Point Algorithm*

Using the pairs of bitangent points as described in the previous subsection, we get an estimate $(\mathbf{R}_0, \mathbf{t}_0)$ of the rigid displacement to superpose S_1 on S_2. In order to find an accurate rigid displacement we have developed an extension of an algorithm called "the Iterative Closest Point algorithm" which was introduced by several researchers [4–6, 26, 36].[4] We sketch the original ICP algorithm, which searches for the rigid displacement $(\mathbf{R}, \mathbf{t})$ which minimizes the energy

$$E(\mathbf{R}, \mathbf{t}) = \sum_{M_i \in S_i} \|\mathbf{R}M_i + \mathbf{t} - ClosestPoint(\mathbf{R}M_i + \mathbf{t})\|^2,$$

where *ClosestPoint* is the function which associates to a space point its closest point on S_2.

The algorithm consists of two iterated steps, each iteration i computing a new estimation $(\mathbf{R}_i, \mathbf{t}_i)$ of the rigid displacement.

1. The first step builds a set $Match_i$ of pairs of points. The construction is the following: for each point M on S_1, a pair (M, N) is added to $Match_i$, where N is the closest point on S_2 to the point $\mathbf{R}_{i-1}M + \mathbf{t}_{i-1}$. To compute the closest point, several methods have been proposed, for example the distance map method [10].

2. The second step is the least-squares evaluation of the rigid displacement $(\mathbf{R}_i, \mathbf{t}_i)$ to superpose the pairs of $Match_i$ (see for example [12] who use a quaternion-based method).

The termination criterion depends on the approach used: the algorithm stops either when (a) the distance between the two surfaces is below a fixed threshold, (b) the variation of the distance between the two surfaces at two successive iterations is below a fixed threshold, or (c) a maximum number of iterations is reached.

The convergence of the algorithm is clearly demonstrated in [8]. Let us define the energy

$$E(\mathbf{R}, \mathbf{t}, Match) = \sum_{M_i \in S_1} \|\mathbf{R}M_i + \mathbf{t} - Match(M_i)\|^2,$$

where *Match* is a function from the 3D space to S_2. The iterative algorithm minimizes this energy E. In step 1, the variables $(\mathbf{R}, \mathbf{t})$ are fixed and E is minimized with respect to *Match*. The function which minimizes E in this case is *ClosestPoint*. In step 2, the variable *Match* is fixed and E is minimized with respect to $(\mathbf{R}, \mathbf{t})$. Hence, at each step, $\mathbf{E}$ decreases. Because E is positive, convergence is guaranteed.

This ICP algorithm is efficient and finds the correct solution when the initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$ of the rigid displacement is "not too bad" and when each point on S_1 has a correspondent on S_2. But in practice, this is often not the case. For example, in our application, as explained in the previous section, the reconstructed surface usually only partially describes the patient's surface and often includes a description of the patient's environment. The next two subsections explain how we deal with these two problems.

2.3. *Working with Incomplete Surfaces*

In step 1 of the iterative algorithm, we map each point of S_1 to a "closest point" on S_2. But when the two surfaces are partially reconstructed, some points on S_1 do not have any homologous point on S_2. Thus, given a point M on S_1, $(\mathbf{R}_{i-1}, \mathbf{t}_{i-1})$, and $ClosestPoint(\mathbf{R}_{i-1}M + \mathbf{t}_{i-1})$, we have to decide whether $(M, ClosestPoint(\mathbf{R}_{i-1}M + \mathbf{t}_{i-1}))$ is a plausible match. This is very important because, if we accept incorrect matches, the found rigid displacement will be biased (and therefore inaccurate), and if we reject correct matches, the algorithm may not converge toward the best solution.

As proposed in [1], we make use of the extended Kalman filter (EKF). This allows us to associate to the six parameters of $(\mathbf{R}_i, \mathbf{t}_i)$ a covariance matrix $\mathbf{S}_i$ and to compute a generalized Mahalanobis distance δ for each pair of matched points (M, N). This generalized Mahalanobis distance, under some assumptions on the noise distributions and some first-order approximations, is a random variable with a χ^2 probability distribution. By consulting a table of values of the χ^2 distribution, it is easy to determine a confidence level ε for δ corresponding to, for example, a 95% probability of having the distance δ less than ε. In this case, we can consider the match (M, N) as *likely* or *plausible* when the inequality $\delta < \varepsilon$ is verified and consider any others as *unlikely* or *unplausible*.

This distinction between plausible and unplausible matches implies a change in the second step of the iterative algorithm. Given $Match_i$, instead of computing the rigid displacement $(\mathbf{R}, \mathbf{t})$ which minimizes the least-squares criterion

$$\sum_{(M,N) \in Match_i} \|\mathbf{R}_iM + \mathbf{t}_i - N\|^2,$$

---

[4] Note that an extension of this algorithm to the nonrigid case can be found in [13] and [30].

we recursively estimate the six parameters of $(\mathbf{R}, \mathbf{t})$, and the associated covariance matrix which minimizes the criterion

$$\sum_{(M,N)\in Match_i \text{ and } (M,N) \text{ is } plausible} (\mathbf{R}_i M$$

$$+ \mathbf{t}_i - N)^t \mathbf{W}_j^{-1} (\mathbf{R}_i M + \mathbf{t}_i - N),$$

where $\mathbf{W}_j$ is a covariance matrix associated with each pair (M, N) which allows us, for example, to increase the importance of high curvature points.

More details about the meaning of "plausible or not" and about the EKF can be found in Appendix C.

2.4. Using the Normal Information to Find the Global Minimum

As is commonly encountered with any minimization algorithm, the ICP algorithm may become trapped in a local minimum. To reduce this problem, we propose in this subsection to make use of the normal information and to define a new criterion to minimize. In our formulation, surface points are no longer 3D points: they become 6D points. Coordinates of a point M on the surface S are (x, y, z, n_x, n_y, n_z), where (n_x, n_y, n_z) is the normal to S at M. For two points $M(x, y, z, n_x, n_y, n_z)$ and $N(x', y', z', n'_x, n'_y, n'_z)$ we define the distance

$$d(M, N) = (\alpha_1(x - x')^2 + \alpha_2(y - y')^2$$

$$+ \alpha_3(z - z')^2 + \alpha_4(n_x - n'_x)^2$$

$$+ \alpha_5(n_y - n'_y)^2 + \alpha_6(n_z - n'_z)^2)^{1/2},$$

where α_i is the inverse of the difference between the maximal and minimal value of the ith coordinate of points in S_2. Using this definition of distance, the closest point to P on S_2 is a compromise between the 3D distance and the difference in normal orientation.[5]

This new definition of the distance between points naturally implies modifications to steps one and two of the ICP algorithm in order to minimize the new energy

$$E(\mathbf{R}, \mathbf{t}) = \sum_{M \in S_1} d((\mathbf{R} M + \mathbf{t}, \mathbf{R} n_1(M)),$$

$$ClosestPoint\_6D((\mathbf{R} M + \mathbf{t}, \mathbf{R} n_1(M)))^2,$$

where $n_1(M)$ is the normal on S_1 at point M and $ClosestPoint\_6D$ is the new 6D closest point function.

In step one, the closest point now has to be computed in 6D space. Because we cannot extend the technique described in [10] (the distance map image would be much too large), we use the kd-tree technique first proposed by Zhang ([36]) for the 3D case. The second step must also be modified: the criterion which defines the best rigid displacement must use the new 6D distance. Otherwise, it is not possible to prove the convergence of our new ICP algorithm as in Subsection 2.2. Hence, the rigid displacement $(\mathbf{R}_i, \mathbf{t}_i)$ is now defined as the minimum of the function

$$f(\mathbf{R}, \mathbf{t}) = \sum_{(M,N)\in Match_i} d(\mathbf{R} M + \mathbf{t}, N)^2,$$

where, the coordinates of the point $\mathbf{R} M + \mathbf{t}$ are $(\mathbf{R} M + \mathbf{t}, \mathbf{R} n_1(M))$.

In practice, we use extended Kalman filters to minimize this new criterion at step 2. Even though it is in nonlinear, the minimization works very well. We have carried out numerous experiments on synthetic data, and the global minimum was always found even with rather large data noise and rather crude initial estimates. Note that this use of extended Kalman filters allows us to compute Mahalanobis distances and to determine if a match is plausible or not as explained in the previous subsection.

In order to try to demonstrate that the ICP algorithm, using the 6D distance, converges more often to the global minimum than the standard ICP algorithm, we conducted the following experiment. We chose S_1 and S_2 to be the same surface. Hence, the resulting transformation should be the identity. We run both the original and the modified algorithm choosing different initial rigid displacements $(\mathbf{R}_0, \mathbf{t}_0)$, at an increasing distance from the identity. The results are reported in Table 1. This shows that our modified algorithm is in practice much less sensitive to the initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$, and more robust to local minima.

2.5. Results

We now present an example application of the framework presented in this section.

First we compute on both the stereo reconstructed surface, and on the MRI surface (Fig. 2), the pairs of bitangent points. We find 598 pairs on the stereo surface and 5000 pairs on the MRI surface (obviously, the desired density of the bitangent pairs is a parameter of the bitangent extraction algorithm). Hence, there are more pairs on the MRI surface than on the stereo surface which makes the algorithm more efficient. The extraction requires about 30 s.[6]

---

[5] Of course, only two parameters are necessary to describe the orientation of the normal (for example the two Euler angles). But we use (n_x, n_y, n_z) because the Euclidean distance better reflects the difference of orientation between the normals (that is not the case with the Euler angles because of the modulo problem) and we can use kd-trees to find the closest point as explained later.

[6] CPU times are given for a DEC–ALPHA workstation.

TABLE 1
Superiority of the Algorithm Using the Surface Normal Information: Quantitative Experiments

σ_r(radians)–σ_t(mm)	0.1–0	0.25–0	0.5–0	0.75–0	1.0–0	1.5–0
points	1000	963	598	293	157	61
points + normals	1000	999	881	596	370	170
σ_r(radians)–σ_t(mm)	0.1–10	0.25–10	0.5–10	0.75–10	1.0–10	1.5–10
points	1000	964	582	286	154	68
points + normals	1000	997	858	567	360	168
σ_r(radian)–σ_t(mm)	0.1–100	0.25–100	0.5–100	0.75–100	1.0–100	1.5–100
points	997	921	512	247	139	52
points + normals	1000	995	834	549	348	152
σ_r(radian)–σ_t(mm)	0.1–500	0.25–500	0.5–500	0.75–500	1.0–500	1.5–500
points	640	576	312	158	85	33
points + normals	906	870	695	456	295	147

Note. We chosen for S_1 and S_2 the same surface: 100 random points chosen on a mannequin head surface. Then we randomly chose rigid displacements, ran both the original and the modified algorithm, and verified if the result was the identity or not. The laws for choosing the rigid displacement are Gaussian laws centered on the identity. For each covariance matrix, we made 1000 trials and we report in the table the number of successes. For example, for a standard deviation of 0.1 radians for the rotation and 500 mm for the translation, the standard ICP algorithm found the correct solution 640 times out of 1000 trials, whereas the result was 906 using the normal information. Another example is for a standard deviation of 1.0 radians for the rotation and 10 mm for the translation: the number of successes are respectively 151 and 360. For more information on Gaussian laws for choosing rotations, see [27]. We note that the standard deviation for rotations is a number between 0 and π.

Using these pairs of bitangent points, we estimate the rigid displacement in about 30 s. Applying this estimate, 80% of the points on the stereo surface have their closest point at a distance lower than 8 mm. This error must be compared with the size of the voxel in the MRI image: 4 mm × 4 mm × 2 mm. Moreover, recall that there are points on the stereo surface which do not a homologous point on the MRI surface.

Using this estimate of the rigid displacement, we run the modified iterative closest point algorithm. The MRI head surface is described by 15000 points and the stereo surface by 10000 points. It takes 20 s. Applying this new rigid displacement, 85% of the points on the stereo surface have their closest point at a distance lower than 3 mm. The average distance between matched points is 1.6 mm. The result is presented in Fig. 3, left.

Because we know the point-to-point correspondences between the MRI head surface and the stereo face surface (this is the result of the registration), and because of each point of the stereo surface we know the gray level from the video image, we can map the video image onto the MRI surface (Fig. 3, right). The fact that the points on the MRI surface have the right gray levels qualitatively demonstrates that the MRI/stereo matching is correct.

Once the MRI head surface has been colorized, we can apply the projective transformation computed as explained
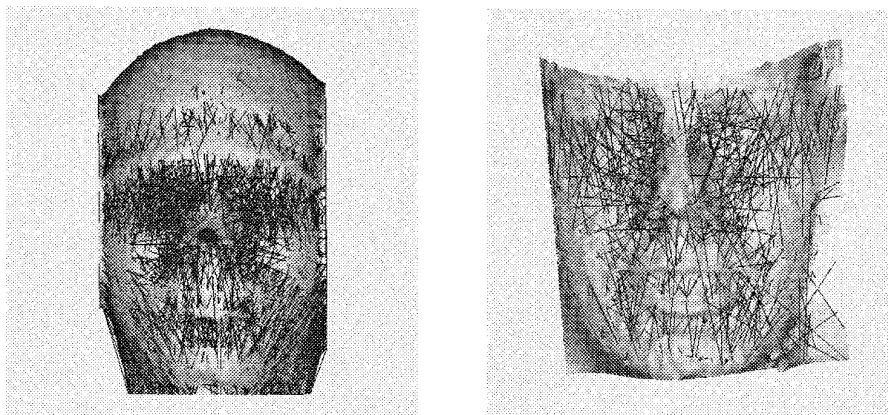

FIG. 2. The bitangent lines computed on the MRI (left) and stereo (right) face surfaces. Note that we selected the lines whose length varies between 2 cm and 10 cm.
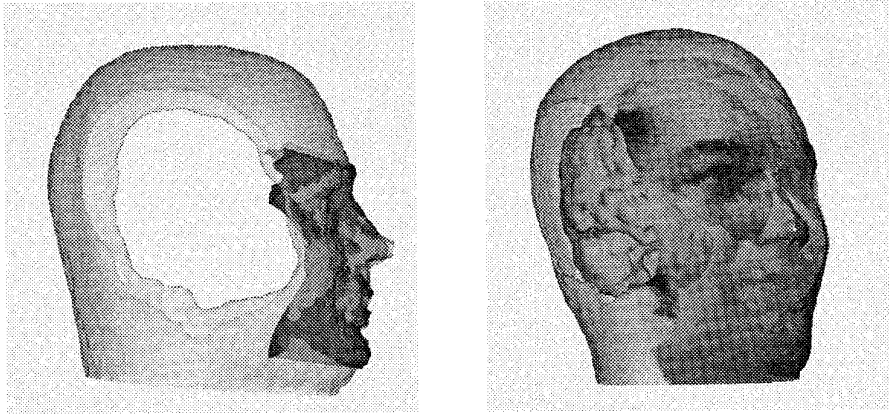
FIG. 3. (**left**) The result of the rigid registration. The MRI surface is clearer and the stereo surface is darker. The alternation dark/clear shows that the registration is quite accurate. (**right**) The colorized MRI head surface obtained by attaching to each matched point the gray level of its corresponding point on the stereo surface. Thanks to the transparency effect, one can observe the brain.

in this paper and get a gray-level 2D image. This image is very similar to the video image and this demonstrates that we actually recover the right projective transformation (Fig. 4).

Finally, we projected the brain onto the video image (see Fig. 5) using the computed projective transformation. In fact, we now have enough geometric and textural parameters to produce a stereo pair of realistic images from a continuous range of viewpoints and provide the surgeon the feeling of seeing inside the patient's head and guide him/her during the operation as explained in [17].

3. 3D–2D PROJECTIVE CURVE REGISTRATION

In this section, we consider the problem of finding the projective transformation between a 3D image of vessels

(for example in the brain or kidney) and a 2D angiography of those vessels (Fig. 8). Two techniques are used to get the 3D images of the vessels. It can either be a MRI image where all the vessels are visible, or it can be a CT image. In the latter case, opaque liquid is injected in a vessel and only those vessels linked to the one injected are visible in the 3D image. The 2D image is a standard radiography and some opaque liquid is also injected in order to make the vessels visible.

An interesting application of the 3D–2D vessel registration is in interventional radiography. A common operation for the radiologist is to introduce a catheter into an artery and guide it toward a lesion within a vessel. The radiologist injects opaque fluid and takes radiographies throughout the operation (which can be quite long) in order to see the vessels and the catheter. The opaque liquid can be

FIG. 4. The left image shows one of the video images. The right image is the gray-level image obtained by applying to the colorized MRI surface the recovered projective transformation. Let us call this image the "simulated image." Note that the white areas come from unmatched points. In order to demonstrate that the video and the simulated images are very similar, we computed the contours $\mathscr{C}$ in the video image: $\mathscr{C}$ is visible in the left picture. Then we superimposed $\mathscr{C}$ int the simulated image. One can observe that $\mathscr{C}$ fits very well with the simulated image. This demonstrates that the recovered projective transformation is correct.
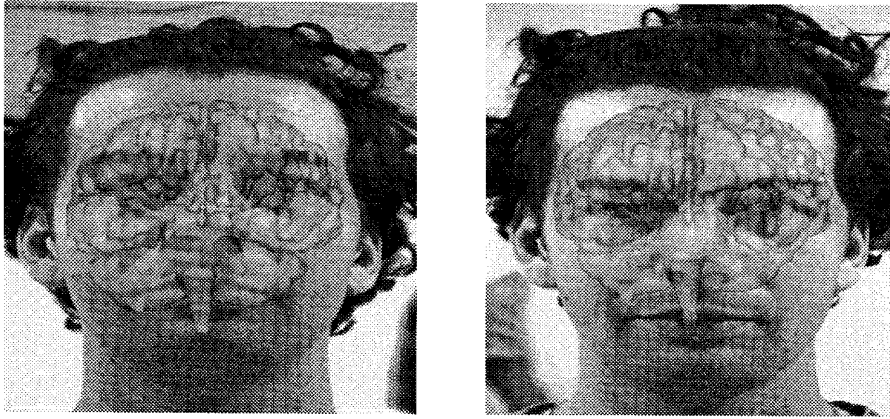
FIG. 5. The projection of the brain in the two video images using the projective transformation computed as explained in this paper. A stereoscopic display could provide the surgeon with the feeling of seeing inside the patient's head. The two presented images cannot be visually fused in this position, because the baseline between the two optical centers of the video cameras was vertical in this experiment: one can notice that the camera of left image was bellow the camera of the right one. This will be corrected in further experiments.

dangerous for the patient, and the X-radiations are noxious for both the radiologist and the patient. Also, understanding the 3D shape of the vessels from two radiographies is a very hard task.

Knowledge of the projective transformation which maps the 3D vessels onto the 2D ones enables us to determine the position of the catheter (which is visible in the radiography) with respect to the 3D image. Hence, the radiologist could visualize his catheter in the 3D image during the operation. This is a significant step toward finding the right path within the vessels. Since this reduces the risk of an incorrect interpretation, the operation becomes safer. Moreover, by reducing the interpretation time, the quantity of injected liquid and radiations can be lowered.

Some recent work ([25]) enables us to represent the vessels as curves described by points in both the 3D and the 2D images. Moreover, spline approximation yields the tangents to these curves [18]. Hence, from a computer vision point of view, the problem is to find the best projective transformation which maps a given 3D curve onto a given 2D curve, the two curves being described by points and their associated tangents. This problem has already been addressed for special curves such as circles or ellipses but not for free-form curves to our knowledge (a hint can be found in [19] for nonplanar parameterized curves).

Note that we know the camera calibration matrix $\mathbf{T}$ corresponding to the 2D image processing.[7] So, it is possible to assume without loss of generality (see Appendix E) that the camera image formation corresponds to perspective projection with center $O = (0, 0, 0)$ and a focal plane

FocalPlane having equation $z = 1$. This means that a 3D point $M = (X, Y, Z)$ is transformed to the point $m = (X/Z, Y/Z, 1)$ which is the intersection between the line OM and the plane *FocalPlane* (see Fig. 6). For clarity, we make this assumption in the sequel of this article. We consider that a point $m = (x, y, 1)$ belonging to the *FocalPlane* is a 2D point with coordinates (x, y). We call the perspective transformation which transforms the 3D point M to the 2D point m, the transformation *Proj*, and we denote this by $m = Proj(M)$. The problem now is to find the rigid displacement $(\mathbf{R}, \mathbf{t})$ such that, for all points M_i on the 3D curve corresponding to the point m_i on the 2D curve, $m_i = Proj(\mathbf{R}m_i + \mathbf{t})$. The projective transformation search for is $Proj\ o\ (\mathbf{R}, \mathbf{t})$.
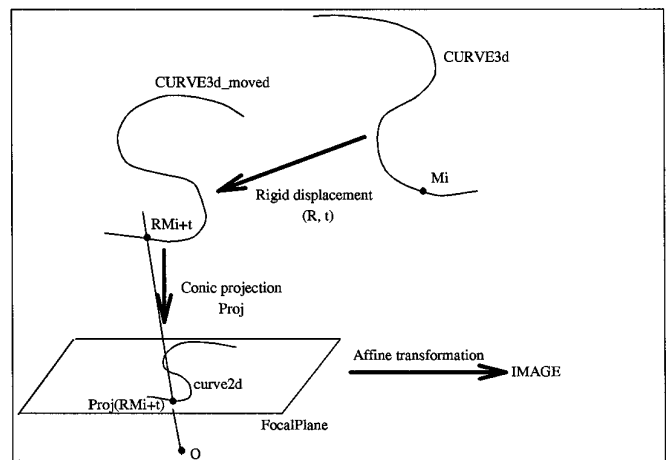

FIG. 6. Each point M_i on the 3D curve is first transformed into $\mathbf{R}M_i + \mathbf{t}$ and then transformed into $Proj(\mathbf{R}M_i + \mathbf{t})$. Finally, an affine transformation of *FocalPlane* provides the image coordinates.

[7] We make the pinhole camera assumption in this paper. This is well justified for our applications.
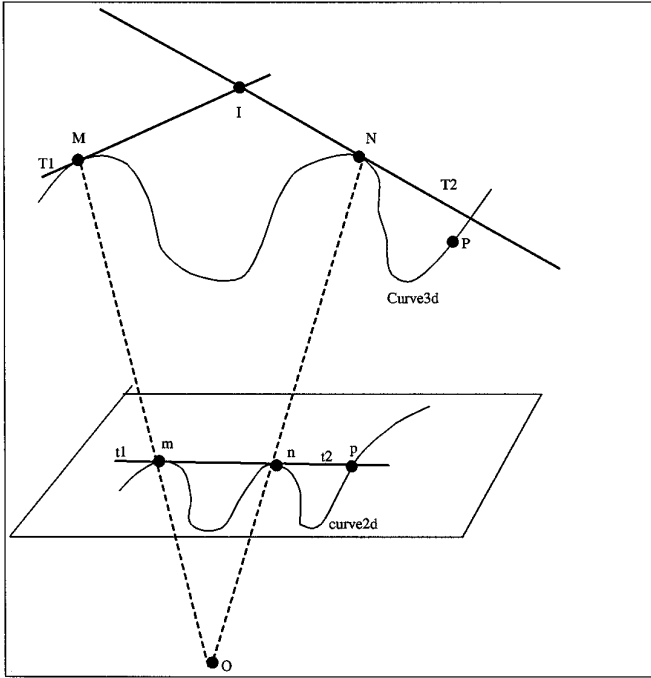
FIG. 7. Illustration of the two properties of curves projection used in this paper. Two points *m* and *n* which share the same tangent lines *l* are projections of two 3D points *M* and *N* such that the tangent lines at these points lie in the same plane *Plane*. Moreover, if *p* is an intersection point between *l* and the 2D curve, then it is the projection of a 3D point *P* which is an intersection point between the 3D curve and *Plane*.

3.1. Finding an Initial Estimate of the Projective Transformation

A well-known result is that knowledge of three points M_1, M_2, M_3 on the 3D object and of the three corresponding 2D points m_1, m_2, m_3 in the 2D image yields four rigid displacements $(\mathbf{R}, \mathbf{t})$ which satisfy $m_i = Proj(\mathbf{R}M_i + \mathbf{t})$ (see for example [23]). Note that there are six equations and six unknowns. Of course, it is not possible to select all the 3D and 2D triplets and to explore the possible combinations in order to find an estimate of the transformation. We must select from the 3D and 2D curves two subsets of triplets which can match each other. In order to reduce the number of potential matches, we use two projective properties of curves (see Fig. 7):

FIRST BITANGENT LINES PROPERTY. *Let c be a 2D curve which is the perspective projection of a 3D curve C (c = Proj(C)). Two points on c which share the same tangent lines are projections of two points on C such that the tangent lines are coplanar.*

Proof. Let M_1 and M_2 be two points on *C*. Let T_1 and T_2 be the attached tangents. Let m_1 and m_2 be the projection of M_1 and M_2. If m_1 and m_2 share the same tangent line, these tangent lines are projections of two 3D tangent lines on

C which lie in the plane defined by (O, m_1, m_2). This proves that T_1 and T_2 are coplanar.

SECOND BITANGENT LINE PROPERTY. *Let t be the common tangent line on c at m_1 and m_2. Let m_3 be an intersection point between t and c. m_3 is the projection of a point M_3 which is an intersection point between the plane defined by (M_1, M_2, T_1, T_2) and the curve C.*

Proof. $O, m_1, m_2, m_3, M_1, M_2, T_1, T_2$ all lie in the same plane *P*. Because M_3 is on the line Om_3, M_3 also lies in *P*.

We use these two properties to find an initial estimate of the projective transformation which maps a given 3D curve *C* onto a given 2D curve *c* as follows:

1. Choose a triplet (m_1, m_2, m_3) of points on the 2D curve *c* such that (a) the tangent lines at point m_1 and m_2 are the same line *l*; and (b) m_2 is an intersection point between *c* and *l*.

2. For each triplet (M_1, M_2, M_3) of points on the 3D curve *C* such that (a) the tangent lines at point M_1 and M_2 lie in the same plane *P*; and (b) M_3 is an intersection point between *C* and *P*,[8] compute the projective transformations which maps (M_1, M_2, M_3) on (m_1, m_2, m_3). Stop if a computed projective transformation is a correct initial estimate.

3. If no solution was found, return to point 1.

More details on this algorithm can be found in Appendix D.1. We note that the complexity is quasi-linear in the number of points describing the curve. It allows us to find an estimate of the projective transformation which maps the 3D curve onto the 2D curve. But, because we need a more accurate transformation, we use this estimate to initialize a two-step minimization algorithm.

3.2. Extension of the ICP Algorithm to 3D–2D Curve Registration

3.2.1. First Extension. The transformation of a 3D point *M* by the estimate of the projective transformation found by the method explained in the previous subsection corresponds to: (i) the application of a rigid displacement $(\mathbf{R}_0, \mathbf{t}_0)$ to *M* and, (ii) the application of the projection *Proj*. In order to get a more accurate projective transformation, we propose to minimize the energy

$$E(\alpha, \beta, \gamma, t_x, t_y, t_z) = \sum_{M_i \in C} \|Proj(\mathbf{R}M_i + \mathbf{t})$$

$$- ClosetPoint(Proj(\mathbf{R}M_i + \mathbf{t}))\|^2,$$

where $(\mathbf{R}, \mathbf{t})$ is the rigid displacement corresponding to the

---

[8] The algorithms to compute the set of triplets are given in Appendix A.

three Euler angles α, β, γ, $\mathbf{t}$ is the translation vector (t_x, t_y, t_z), and where *ClosestPoint* is the function which associates with each point m in *FocalPlane* its closest point on c.

As in the previous section, the algorithm consists of two iterated steps, each iteration j computing a new estimation $(\mathbf{R}_j, \mathbf{t}_j)$ of the rigid displacement:

1. Build a set of pair of points *Match$_j$* by attaching to each point M_i in C the closest point m_i on c to the point $Proj(\mathbf{R}_{j-1}M_i + \mathbf{t}_{j-1})$.

2. Compute the best rigid displacement which minimizes

$$\sum_{M_i \in C} \|Proj(\mathbf{R}M_i + \mathbf{t}) - m_i\|^2.$$

Using the same kind of demonstration as the convergence demonstration of the ICP algorithm presented in Subsection 2.2, it is straightforward to prove that the algorithm converges and minimizes the energy E. Moreover, because we also use extended Kalman filters to perform the minimization at step 2 of the algorithm, we get the covariance matrix associated to the rigid displacement $(\mathbf{R}_{j-1}, \mathbf{t}_{j-1})$. Hence, as in the 3D–3D rigid case, we can compute for each pair in *Match$_j$* a Mahalanobis distance and decide whether a pair in *Match$_j$* is a plausible match or not. This allows us to deal efficiently with the outliers problem. Note that this problem is for example crucial for the 3D–2D brain vessel registration problem because all the vessels are visible in the MRI image, whereas only the injected ones are visible in the radiography.

3.2.2. *Extension Using the Tangent Information.*   As for the 3D–3D rigid registration problem, the exploitation of tangent information enhances the convergence of the 3D–2D ICP algorithm. In this case, the points on the 2D curves are no longer 2D points. They are 4D points (x, y, tan_x, tan_y), where (x, y) are the classical spatial coordinates and (tan_x, tan_y) are the coordinates of the normalized tangent vector to the curve. Hence, given two points m_1 and m_2 with coordinates $(x_1, y_1, tan\_x_1, tan\_y_1)$ and $(x_2, y_2, tan\_x_2, tan\_y_2)$, we can define the distance

$$d(m_1, m_2) = (\alpha_1(x_1 - x_2)^2 + \alpha_2(y_1 - y_2)^2$$
$$+ \alpha_3(tan\_x_1 - tan\_x_2)^2 + \alpha_4(tan\_y_1 - tan\_y_2)^2)^{1/2},$$

where α_i is the inverse of the difference between the maximal and minimal value of the ith coordinate of points in c.

In this case, the criterion corresponding to this new distance is

$$E(\alpha, \beta, \gamma, t_x, t_y, t_z) = \sum_{M_i \in C} d(\mathbf{R}M_i + \mathbf{t}), Proj\_tangent(\mathbf{R}T_i)),$$

$$ClosestPoint\_4D((Proj(\mathbf{R}M_i + \mathbf{t}), Proj\_tangent(\mathbf{R}T_i))))^2,$$

where T_i is the normalized tangent vector to C at point M_i and *Proj_tangent* is the function which associates with the normalized tangent vector $\mathbf{R}T_i$ at point $\mathbf{R}M_i + \mathbf{t}$, the normalized tangent vector to the projected curve at point $Proj(\mathbf{R}M_i + \mathbf{t})$. The modifications to the previous extended ICP algorithm to minimize this new criterion are straightforward and similar to the modifications in the 3D–3D rigid case. In step 1, the closest points are computed in 4D space using the kd-tree technique, and in step 2 the 4D criterion corresponding to the new 4D distance is

$$\sum_{(M_i, m_i) \in Match_j} d(Proj(\mathbf{R}M_i + \mathbf{t}), m_i)^2.$$

As in the 3D–3D case, experiments demonstrate that the use of tangent information makes the minimization algorithm less sensitive to its initialization.

3.3.   *Results*

3.3.1.   *3D Morphometer Data.*   We present an application of this framework to the data shown in Fig. 8. The 3D data have been acquired by the "3D morphometer".

Figure 9 (left) shows the initial estimate. One can observe that it is quite different from the one shown in Fig. 8 (left). Indeed, different projections of the same 3D curve can produce very different images. This demonstrates that the search for the initial transformation (Subsection 3.1) is crucial. Figure 9 (right) shows the result of the minimization process presented in Subsection 3.2 from this initial estimate. The computation time is 20 s on a DEC alpha workstation. For matched points, the average distance is 6.8 pixels for the initial estimate and 0.83 pixel for the final registration.

It is difficult to estimate the accuracy of the registration with this measure of the error. An improved quantitative measure would require two radiographies taken with a 90° angle. This would allow us to measure the z error. We plan to conduct such experiments in the near future.

3.3.2.   *3D MR Aniography Data.*   We also conducted some experiments to match 3D MR angiographies onto 2D X-ray angiographies. This scheme has a large potential use in practice because the acquisition of MR angiography is noninvasive. This is the reason why we treat the vessels as 3D curves. Indeed, the vessels generally appear thinner in MR images than in 2D X-rays. Even if the skeleton is not invariant under projective transformation, this approximation works well in practice because the vessels are thin enough.

A stereotactic frame was attached to the head of the patient during the acquisition of the images. This allowed us to compare quantitatively the projective transformation $\mathbf{T}_{frame}$ found using the stereotactic frame and the transfor-
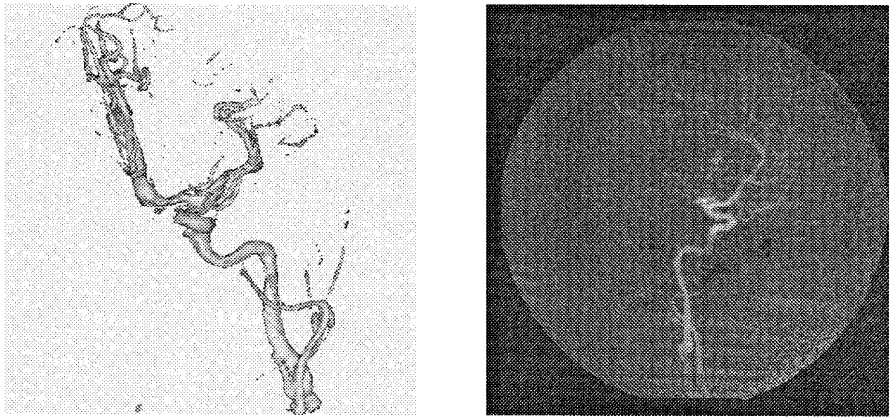
FIG. 8. (**left**) The 3D model of the vessels. It was extracted from a 3D image acquired by the "3D morphometer" of General Electric Medical System Europe. (**right**) A radiography of the vessels (an angiogram). Our goal is to find the point to point correspondence between the left and right objects.

mation $\mathbf{T}_{frameless}$ found without the frame as described in this section, using only the vessels visible in the images.

In order to compare $\mathbf{T}_{frame}$ and $\mathbf{T}_{frameless}$, we define an error measure as follows. For each point M in the 3D volume of interest, we apply T_{frame} to M. We get a 2D point m. The inverse of m for the transformation $\mathbf{T}_{frameless}$ is a set of points M' forming a line L. We define the error associated with the point M as the distance between M and the line L.

Typically, we get an average error of 0.82 mm, a maximal error of 2.04 mm and a minimal error of .0001 mm. This error is close to the uncertainty attached to the estimation of $\mathbf{T}_{frame}$. Hence, from an accuracy point of view the "frame" and "frameless" techniques are similar. The enormous advantage of the frameless technique is that it is noninvasive and permits comparison between images taken with a long time interval. On the other hand, the

frame technique is probably more robust. We believe that using two 2D angiographies, a frontal one and a lateral one, the frameless technique could be robust enough for clinical use. Note that when several radiographies are available, and assuming that the relative acquisition positions are known, it is then straightforward to define a new criterion being the sum of the individual criterions define above for each image and therefore to take into account both radiographies.

4. 3D–2D PROJECTIVE SURFACE REGISTRATION

Even if a vessel is not visible both in the 3D image and in the 2D radiography, it can be useful to find the projective transformation which maps the 3D image onto the 2D radiography. This is for example the case in radiotherapy (as explained in [28]), where one problem is to link the
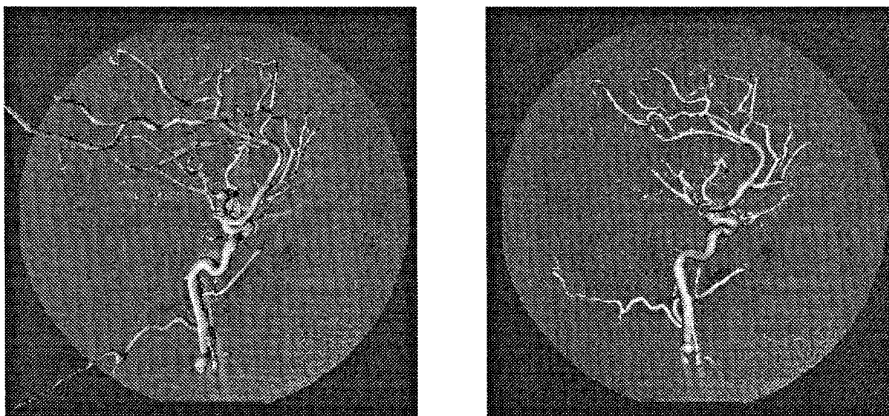

FIG. 9. (**left**) The initial estimate of the transformation. (**right**) The result of the minimization process. One can observe that the radiography vessels are no longer visible which shows that the 3D model is quasi-perfectly projected on them. This demonstrates the accuracy of the found projective transformation.
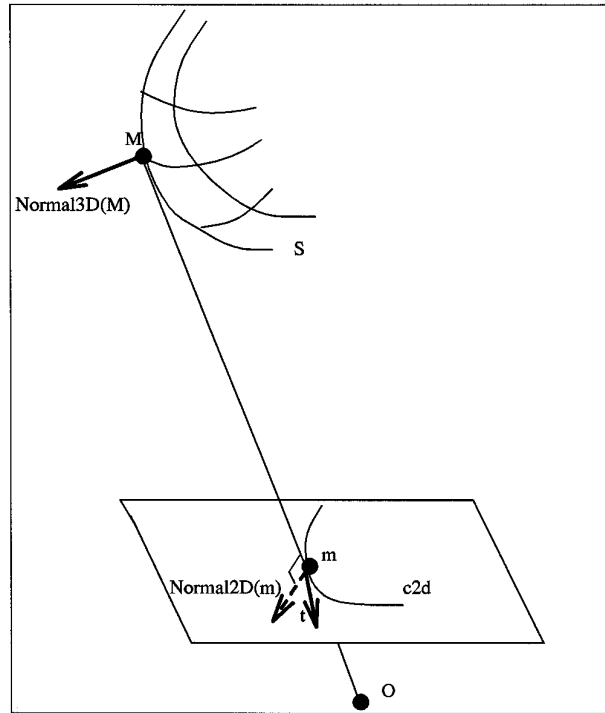
FIG. 10. The illustration of the fundamental property of occluding controur formation. The projection of the point M is m. m lies on the occluding contour and the tangent to this contour is **t**. The normal vector to S at point M is equal to the normal to the plane defined by $(O, m, \mathbf{t})$.

MRI image to the actual position of the patient in order to guide the radiation sources. We recall that the usual practice to complete this task is to screw a stereotactic frame in the skull of the patient!

The approach described in this section resembles the approach introduced by Lavallee in [21]. We find the position of the 3D object so that its projection corresponds to the occluding contours detected in the 2D image. But Lavallée's approach assumes that it is possible to determine globally the interior and the exterior of the occluding contour. This is often a hard task in practice. The use of the vector normals allows us to avoid this problem. Moreover, we do not make any assumption about the initial position of the 3D object. Ponce also used the occluding contour in [20] to find the pose of 3D objects, but his scheme requires second-order derivatives, which can significantly reduce the robustness of the approach.

4.1. Fundamental Property of Occluding Contour

The basic idea of our approach is to use a

FUNDAMENTAL PROPERTY OF OCCLUDING CONTOUR (see Fig. 10). *If a point M on a 3D surface S is so that projection point $m = Proj(M)$ lies on the occluding contour c, then the normal vector $Normal_{3D}(M)$ to S at point M is equal to the normal vector **n** of the plane P defined by $(m, O, \mathbf{t})$, where **t** is the tangent vector to the occluding contour at point m.*

We call $Normal_{2D}$ the function which associates the normal **n** to the point m and we note $\mathbf{n} = Normal_{2D}(m)$.

4.2. Finding an Initial Estimate of the Projective Transformation

As in the previous section, we use two properties of surface projection in order to select independently on the 3D surface, and on the occluding contour, triplets of points which can correspond. By exploring possible associations of 2D and 3D triplets and verifying them, we will find an estimate of the required projective transformation. These properties are:

PROPERTY 1.[9] *Two 2D points m_1 and m_2 on the occluding contour which share the same tangent line are projections of two points M_1 and M_2 on the surface which share the same tangent plane.*

Proof. All tangent lines to the surface at points M_1 and M_2 lie in the plane (O, m_1, m_2).

PROPERTY 2.[10] *Let m_1 and m_2 be two 2D points on the occluding contour and let M_1 and M_2 be two points on the 3D surface such that $m_1 = Proj(M_1)$ and $m_2 = Proj(M_2)$. The angle between $Normal_{2D}(m_1)$ and $Normal_{2D}(m_2)$ is*

[9] Note that this property has already been used by Zisserman [15] to find the pose of symmetric objects.

[10] Note that property 1 is a consequence of property 2.

equal to the angle between the normals $Normal_{3D}(M_1)$ and $Normal_{3D}(M_2)$ on the surface at points M_1 and M_2.

Proof. The two planes defined respectively by $(m_i, Normal_{2D}(m_i))$ and $(M_i, Normal_{3D}(M_i))$ are identical.

Hence, to find the initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$ of the rigid displacement so that *Proj o* $(\mathbf{R}_0, \mathbf{t}_0)$ is the projective transformation which transforms S with the 2D occluding contour c, we proceed as follows:

1. Choose (m_1, m_2, m_3) on the 2D curve c such that: (a) m_1 and m_2 share the same tangent line; and (b) the angle α between $Normal_{2D}(m_1)$ and $Normal_{2D}(m_3)$ is as close as possible to π.

2. For each triplet (M_1, M_2, M_3) on the 3D surface S such that (a) M_1 and M_2 share the same tangent plane; and (b) the angle between $Normal_{3D}(M_1)$ and $Normal_{3D}(M_3)$ is α, compute the projective transformations which maps (M_1, M_2, M_3) onto (m_1, m_2, m_3). Stop if a computed projective transformation is a correct initial estimate.[11]

3. If no solution was found, return to point 1.

More detail about this algorithm (in particular about the way in which we deal with discretization error) can be found in Appendix D.2. We note that the complexity is quasi-linear in the number of points describing the surface. As in the previous section, this algorithm yields an estimate of the searched projective transformation and we use it to initialize an energy minimization algorithm and get an accurate transformation.

4.3. Extension of the ICP Algorithm to 3D–2D Surface Registration

The previous estimate of the projective transformation corresponds to, first, the application of a rigid displacement $(\mathbf{R}_0, \mathbf{t}_0)$ and, then, the application of the projection *Proj*. In order to get a more accurate projective transformation, we propose a distance between the points on the surface and the points on the occluding contour and minimize an appropriate energy function.

Let $m = (x, y)$ be a point on c and let $M = (X, Y, Z)$ be a point on S. We define the 5D distance

$$d(m, M) = (\alpha_1(x - X/Z)^2 + \alpha_2(y - Y/Z)^2$$
$$+ (Normal_{2D}(m) - Normal_{3D}(M))^2)^{1/2},$$

where α_1 and α_2 are the inverse of the difference between the maximum and the minimum values of respectively the

x- and y-coordinates of points on c. The energy corresponding to this new distance is

$$E(\alpha, \beta, \gamma, t_x, t_y, t_z, Match) = \sum_{m_i \in c} d(m_i, \mathbf{R}Match(m_i) + \mathbf{t})^2,$$

where *Match* is a function which associates a point on S with each point on c. This energy measures how well the "fundamental property of occluding contour" (Subsection 4.1) is verified.

As in the two previous sections, we use a two-step algorithm to minimize this energy. This algorithm can be understood as an extension of the ICP algorithm. The first step is the minimization of E with respect to *Match*, assuming that α, β, γ, t_x, t_y, t_z are fixed. The second step is the minimization of E with respect to α, β, γ, t_x, t_y, t_z assuming that *Match* is fixed. We now describe the algorithm which consists of two iterated steps, each iteration j computing a new rigid displacement $(\mathbf{R}_j, \mathbf{t}_j)$:

1. We associate with each point m_i on c a point M_i. To do so, we first compute the set

$CloseNormal(m_i)$

$= \{M \in S/Angle(\mathbf{R}_{j-1}^t Normal_{2D}(m_i), Normal_{3D}(M)) < \varepsilon\},$

where ε is a given threshold.[12] Next we compute the point N which minimizes $d(m_i, \mathbf{R}_{j-1}M + \mathbf{t}_{j-1})$ for all M in *CloseNormal*. In order to ensure that E decreases at this step 1 (this is the condition required to prove the convergence of the algorithm), we compare $d(m_i, \mathbf{R}_{j-1}N + \mathbf{t}_{j-1})$ and $d(m_i, \mathbf{R}_{j-1}M_i\_Prev + \mathbf{t}_{j-1})$, where M_i_Prev is the point associated with m_i in the previous iteration.[13] If $d(m_i, \mathbf{R}_{j-1}N + \mathbf{t}_{j-1})$ is smaller than $d(m_i, \mathbf{R}_{j-1}M_i\_Prev + \mathbf{t}_{j-1})$, the matched point M_i becomes N, otherwise it remains as M_i_Prev.

2. Compute the rigid displacement $(\mathbf{R}_j, \mathbf{t}_j)$ which minimizes

$$\sum_{m_i \in c} d(m_i, \mathbf{R}M_i + \mathbf{t})^2.$$

The demonstration of the convergence of this algorithm is straightforward, since at each step E decreases. Moreover, because we use extended Kalman filters to perform the minimization of step 2, we get the covariance matrix corresponding to the six parameters of $(\mathbf{R}_j, \mathbf{t}_j)$. Hence, as in the two previous sections, we can compute a Mahalanobis distance and deal with points on c which have no correspondent on S by performing χ^2 tests (see Appendix C for

---

[11] A technique for verifying the hypothesis is proposed in [3], using the surface curvature, and making the affine camera assumption. We note that we use only the surface normals and that our camera model is the pinhole one (see Appendix D.2).

[12] Note that the computation of this set is fast because we compute, as a preprocessing, a 3D kd-tree on the normals to the surface S.

[13] The convention is that $d(m_i, \mathbf{R}_0 M_i\_Prev + \mathbf{t}_0)$ is infinity.
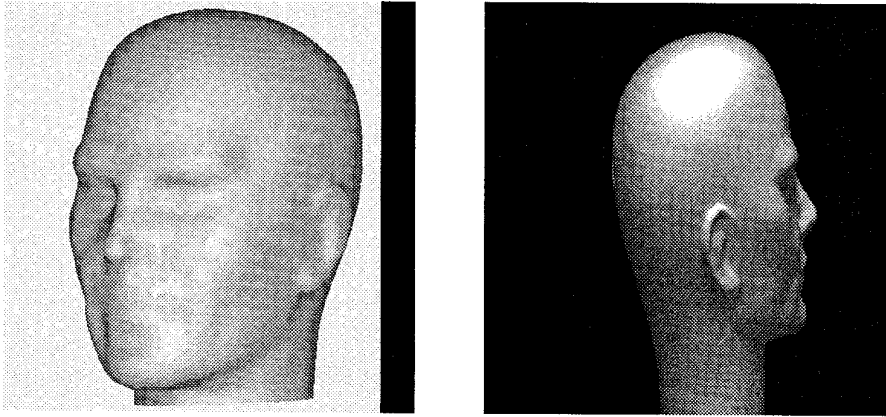
FIG. 11. (**left**) The 3D model of the mannequin head. It was extracted from a 3D CT-scan image. Thanks are due to Cannes Hospital. (**right**) A video picture of the mannequin head. Our goal is to find the point-to-point correspondence between the left and the right objects.

more details). This allows us to find an accurate projective transformation which transforms the surface S with occluding contour c, discarding outliers.

4.4. Results

We now present an application of the framework presented in this section for surface registration using occluding contours.

Figure 11 presents the data to register: a 3D model of a mannequin head extracted from a 3D CT-scan image (left), and a 2D video image of this mannequin head (right). Figure 12 (left) shows the initial estimate, and Fig. 12 (right) shows the result of the minimization stage. The CPU time is 10 s on a DEC alpha workstation.

For the initial estimate, the average error is 2.1 pixels for spatial distance and 2 degrees for the difference in normal orientation. Even if some points are rejected be-

cause they are considered as outliers (they will be considered later during the iterative process), this shows that the initial estimate is already quite good. For the final transformation the average error is 0.76 pixels and 0.17 degrees which demonstrates that it is accurate. We plan to attach a stereotactic frame to the mannequin head and perform the experiment again in order to have a reference measure of the error and then try to better demonstrate the accuracy of the recovered transformation. Note that when the relative positions of the cameras are known, it is possible to integrate into our framework several camera views. The criterion is, in this case, the sum for each image of the criterion defined in the previous subsection.

5. THE NONCALIBRATED CASE

In the two previous sections, we assumed that we knew the calibration matrix attached to the camera. Even if this
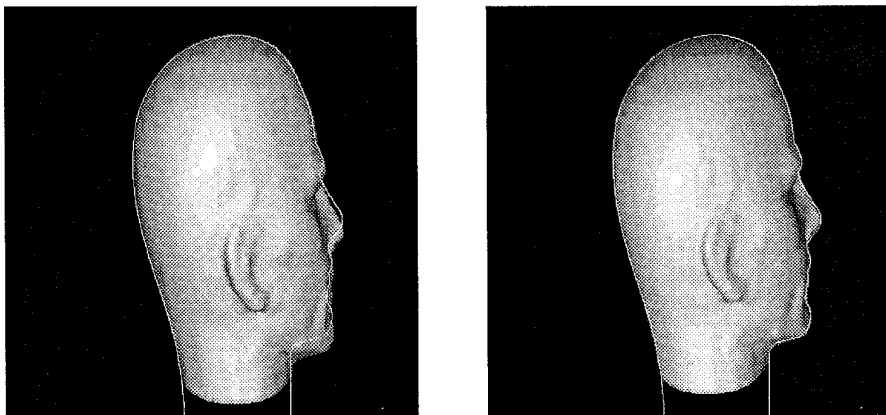

FIG. 12. (**left**) The initial estimate of the transformation: we projected the 3D model using the initial transformation. The contour superimposed in this image is the contour extracted in the video image. (**right**) The final transformation: we projected the 3D model using the transformation found after minimization. You can observe that the contour which comes from the video image exactly corresponds to the contours of the 3D model. This demonstrates that we recovered the correct projective transformation.

seems to be a reasonable assumption because in practice it is possible to calibrate a camera (see for example [1]), or because it is quite often possible to know the calibration parameters for a radiology system, we now explain how to proceed when we do not know the calibration matrix.[14] It is obvious that the problem is harder. Since five calibration parameters are unknown, the problem is to find *all the eleven parameters of the projective transformation.*

5.1. *Finding an Initial Estimate of the Projective Transformation*

We do not (yet) have a satisfactory automatic solution for this problem. In practice, we find the estimate manually. Using an appropriate interface, it is possible to select interactively pairs of corresponding points on the 3D object and the 2D object. Selecting at least six pairs of corresponding points, it is possible, with a least-squares minimization, to determine an initial estimate.

5.2. *Finding an Accurate Projective Transformation*

In practice, there are two cases where we are provided with an estimate of the best projective transformation, and we want to find the best one with respect to the eleven parameters. This is the case when the estimate has been found manually as explained in the previous subsection. But this problem also arises when the camera calibration is not very accurate, a common occurrence in practice.

In such cases, we minimize the same criteria as the ones described in the two previous sections. But instead of minimizing them with respect to six parameters, we minimize them with respect to the eleven parameters of the projective transformation. The minimization algorithms are almost the same. We observe that these 11D minimizations are more sensitive to the initialization conditions than the 6D case. But, because our initial estimates are not too crude, it works well in practice. This allows us to find an accurate projective transformation even if the camera calibration is not perfect, or when the initial estimate has been found interactively.

5.3. *Results*

We have a 3D CT-scan image of a skull and a 2D X-Ray image of it (Fig. 13, right). We do not know the calibration parameters of the radiography system. We first extract the skull surface in the 3D image (Fig. 13, left). Then, we design six pairs of corresponding points on the 3D skull

surface and the X-ray image. This yields the estimate of the projective transformation (Fig. 14, left). The average error is 1.5 pixels for spatial distance and 1.9 degrees for the difference in normal orientation. Then we run the minimization algorithm to recover accurately the 11 parameters of the projective transformation. The average error for the resulting transformation is 0.79 pixels and 0.7 degrees which demonstrates that it is accurate.

6. CONCLUSION

We presented a unified framework to perform 3D–2D projective registration. We first showed how to use the bitangent lines or the bitangent planes to find an estimate of the transformation. Then we proposed definitions of the distance between the 3D object and the 2D image and we presented extensions of the iterative closest point algorithm to minimize this distance dealing with occlusion in an rigorous way.

6.1. *Validation*

Although we presented results on real data, a lot of work still needs to be done. Because the techniques described in this article are related to medical operations, it is especially important to perform a rigorous validation. Two properties must be established precisely: *robustness* and *accuracy.* In the presented framework, we first segment the data, then find an initial estimate of the projective transformation and finally we perform an energy minimization.

The segmentation stage is obviously very important. Both robustness and accuracy depend on it. We did not detail this stage in this paper, more details can be found in [24, 25]. In practice, it works well and the registration remains robust thanks to our treatment of the outliers.

Our search for an initial estimate only involves first-order derivatives and has reasonable algorithmic complexity. We could have chosen to compute higher order derivatives to achieve a lower computational complexity or to compute lower order derivatives to achieve a better robustness. We made the choice presented in this article because it seems to us, after experiments, that it is a good trade-off between robustness and complexity (see [15] for an interesting discussion on these issues). In practice, the different thresholds have been easily chosen and the method is not sensitive to their precise values. But we will not discuss this point here. In practice, an appropriate user interface allows the user to design interactively some corresponding points on the objects to register. Hence, even if an initial estimate is proposed automatically as described in this paper, it can be corrected interactively if it is wrong. Note that the initial estimate does not need to be accurate because it is just an initialization of the minimization stage.

The robustness of the minimization stage is not easy to

---

[14] We do note here about the 3D–3D registration problem addressed in Section 2. But note that it is possible to perform the stereo with noncalibrated cameras. In this case, the reconstructed surface is known up to a 3D–3D projective transformation. Then the corresponding registration problem would be to find the 3D–3D projective transformation (15 parameters) to superpose the two surfaces.
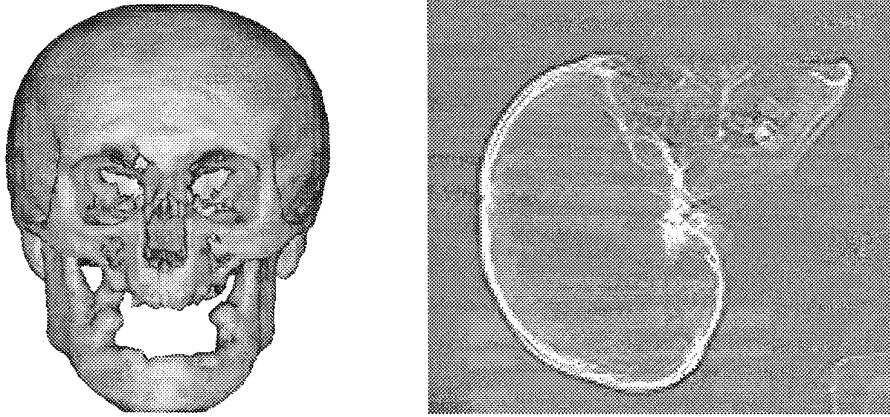
FIG. 13. (**left**) The skull surface extracted from the 3D CT-scan image. (**right**) The skull X-ray image. The goal is to recover the 3D–2D projective transformation (11 parameters) mapping the 3D surface (left) onto the 2D image (right), without any calibration information.

check. It is possible to perform statistics. One can add some noise to the objects to register, perturb the correct solution, run the algorithm, and check if it converges toward the correct solution. Such experiments were presented in Table 1. The next stage is to perform the experiments on a larger data set and it will take a long time. Checking the accuracy is difficult but important too. The best way to do it is probably to compare the automatic registration with the one obtained with a stereotactic frame. Such an experiment was presented in Subsection 3.3.2.

Even if the experiments presented in this paper seem to demonstrate the robustness and accuracy of the techniques (at least in the calibrated case), we believe that for maximal security, it is better to use more than one sensor when possible. For neuradiology, it would be better to use two X-ray views as noted in Subsection 3.3.2. For radiother-

apy, it would be useful to combine the techniques described in Sections 2 and 4. Indeed, the first one is well adapted for frontal views and the second one for lateral views. Note that it is not difficult to combine different viewpoints with the different techniques described in this paper since the defined criterions can easily be added together, as noted in Subsection 3.3.2.

6.2. Future Work

We believe that the techniques described in this paper offer different possible and interesting extensions.

First, we would like to compare from a practical point of view the techniques described in this paper with respect to the techniques presented by others [9, 17, 21, 22, 28, 29, 35]. For example, we plan to develop new procedures to validate and compare rigorously the accuracy of these techniques.
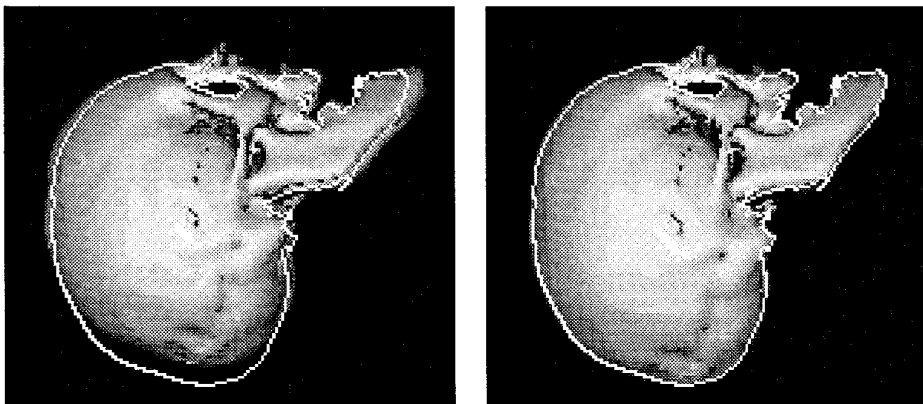

FIG. 14. (**left**) The transformation found designing six pairs of corresponding points on the 3D surface and the 2D image. (**right**) The result of the minimization. You can observe that the contour which comes from the X-ray image exactly corresponds to the contours of the 3D model (except small areas where the contour extraction is not perfect and which are considered as outliers). This demonstrates that we recovered the right projective transformation.

Then, we wish to validate our approach on a larger scale. At least for the vessel registration problem, we plan a clinical validation in collaboration with General Electric Medical Systems Europe and a hospital. We will use a frontal and a lateral view for these experiments.

We believe that it should be possible to perform real-time tracking of the patient and enable the surgeon to move either the patient or the 2D sensor. Indeed, for tracking we need only to correct a projective transformation which is quite close to the right solution as explained in Subsection 5.2. Hence, because the initialization would be good, it should be possible to use the extensions of the iterative closest point algorithm presented in this paper with just a few points without local minimum problem and to get fast convergence for the eleven parameters of the projective transformation.

Finally, we would like to develop a new 3D–2D registration scheme, for example extending some ideas presented by Fua in [14] and try to better understand the 3D–2D registration problem when the 3D object is deformable. This problem arises, for example, in the processing of X-ray mammographies.

APPENDIX A

Computing Special Triplets of Point

A.1. *Computing the Bitangent Planes on a Surface and the Bitangent Line on a 2D Curve*

We present in this appendix our technique for computing, given a surface S described by points and attached normals, the pairs of points sharing the same tangent plane. We do not present the details of the technique for computing the bitangent lines on a 2D curve because it is the same problem in 2D space.

As a preprocessing, we first compute a kd-tree *Tree* based on the Euclidean coordinates of the normal vectors. Then for each point M, we compute the set

$$CloseNormal(M) = \{N \in S / Angel(Normal\_3D(M),$$

$$Normal\_3D(N)) < \varepsilon_M\},$$

where $Normal\_3D(M)$ is the normal to S at point M, *angle* is the function which returns the angle between two normalized vectors, and ε_M is a threshold automatically computed as explained later. For each point M' in *CloseNormal*(M), let P (resp. P') be the tangent plane at point M (resp. M') and let m (resp. m') be the orthogonal projection of M on P (resp. P'). We decide that M and M' are bitangent points if the $(\widehat{MM', Mm'})$ and $(\widehat{M'M, M'm})$ are both smaller than ε_M.

The key parameter of the algorithm is ε_M. This threshold angle must depend on the local curvature of the surface and on the local density of the discretization. In practice, we choose ε_M as the average angle between the normal at point M and the normal at the four closest points in S to M. In order to control the final density of the set *Bitangent*(S), we introduce a parameter δ: we add in *Bitangent*(S) a new bitangent pair (M, M') only if there is no pair (Q, Q') in *Bitangent*(S) so that $(d(M, Q) < \delta)$ *and* $d(M', Q') < \delta)$ or $(d(M, Q') < \delta)$ *and* $d(M', Q) < \delta)$. We use a hash table for performing this test.

The complexity of the technique is $O(p \times n)$, where p is the maximum number of points in *CloseNormal*(M) and n is the number of points describing the surface. In practice, because of the way we compute ε_M, the more the discretization of S is dense, the more ε_M is small. Hence, in practice, p depends only on the "complexity" of the surface[15] and the technique is quasi-linear. Of course, we do not pretend that this analysis is rigorous: we just try to explain why the algorithm is efficient in practice.

A.2. *Computing the Coplanar Tangents on a 3D Curve*

The problem is here to compute, given a 3D curve described by points and attached tangents, the pairs of points such that the tangent lines are coplanar. The technique is very simple. A set of m planes is associated with each tangent line. It is just a uniform sampling of the set of planes containing the tangent line.

Hence, the problem is the same as in Appendix A.1: computing the bitangent planes on a surface. Because m is a constant number (typically 20), the complexity is also quasi-linear.

A.3. *Computing the Intersections between a Set of 2D Lines and a 2D Curve and between a Set of Planes and a 3D Curve*

The problem is, given a set of 2D lines and a 2D curve described by points, to compute the intersection points between the lines and the curve. We first tessellate the plane using squares of edge size d. Hence, each point of the curve belong to a square. For each nonempty square *Square*, we compute the set *CloseLine* of lines which are at a distance below $d \times \sqrt{2} + \delta$ from the square center. Then, exploring all the curve points in *Square* and all the lines in *CloseLine*, we simply retain the set of pairs (M, l) such that M is a point in *Square* and l is a line in *CloseLine* and the distance between M and l is smaller then δ. We choose δ as the average distance between two consecutive points on the curve.

The complexity of the technique is in the worse case $O(p \times n)$, where p is the number of lines and n the number

---

[15] Clearly, we assume that the surface is not flat but smooth. This is a reasonable assumption for anatomical surfaces.

of points describing the curve. So, in the worst case the plane tessellation is unuseful. But in practice, it makes the computation much faster and the calculation time is acceptable.

We do not present the details for computing the intersections between a set of planes and a 3D curve. Indeed, it is the same problem as the previous one in 3D space.

APPENDIX B

Finding the Initial Estimate of the 3D–3D Rigid Registration

B.1. *The Algorithm*

In practice, two corresponding pairs of bitangent points cannot be exactly superposed because of the point discretization error and because of the error in the normal computation. Moreover, only a part of the reconstructed surface S_1 may be superposed on the patient's surface extracted from the MRI image S_2. Indeed, in S_1 the patient's surface is not complete, and there can be a part with corresponds to the reconstruction of the patient's environment. So, in practice, we first compute on S_1 and S_2 the two sets of pairs of bitangent points $Bitangent(S_1)$ and $Bitangent(S_2)$. Then, we sort the set $Bitangent(S_2)$ on the distance between the two bitangent points. This way, given a pair P_1 in $Bitangent(S_1)$, we can very quickly compute the set

$$CloseDistance(P_1)$$
$$= \{P_2 \in Bitangent(S_2)/|d(P_1) - d(P_2)| < \Delta\},$$

where Δ is a given threshold.

Then we apply the following algorithm:

1. We randomly choose a pair P_1 in $Bitangent(S_1)$

2. We compute the set $CloseDistance(P_1)$

3. For each pair P_2 in $CloseDistance(P_1)$, we compute the two rigid displacements $\mathscr{D}$ and $\mathscr{D}'$ which correspond to the superposition of P_1 on P_2. For each of these two rigid displacements, we now estimate the number of points on the transformed surface $\mathbf{R}S_1 + \mathbf{t}$[16] which have their closest point on S_2 below a given distance δ to check whether $\mathscr{D}$ or $\mathscr{D}'$ reaches our termination criterion. First, we randomly choose a subset S_1' of points on S_1. Then, for each point P in S_1', we compute the closest point Q to $\mathbf{R}P + \mathbf{t}$ on S_2. Let m be the number of points P such that $\|PQ\| < \delta$. If the ratio $m/|S_1'|$ is larger than ρ ($0 < \rho < 1$), we decide that $(\mathbf{R}, \mathbf{t})$ is a good estimate of the rigid displacements which superposes S_1 on S_2 and we stop the algorithm.

4. If no solution has been found, we return to point 1.

[16] A rigid displacement $(\mathbf{R}, \mathbf{t})$ maps each point M to $\mathbf{R}M + \mathbf{t}$ where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ a translation vector.

The parameter δ is the estimation of the maximal distance between one point on the surface S_1 and its closest point on S_2 for a good estimate of the best rigid registration (δ depend on the noise level). ρ is the estimation of the ratio of the number of points on S_1 which have a correspondent on S_2, divided by the number of points on S_1 (ρ depends on the occlusion level).

B.2. *Complexity Analysis*

The goal of this subsection is not to provide a very rigorous analysis of the algorithm complexity but just to try to explain "why it works."

First, if the surface does not present too many flat areas, the complexity of the bitangent calculation and the number of bitangents is quasi-linear in the number of points on the surface (see Appendix A.1).

Moreover, the number of iterations of the algorithm is relatively small because we just have to try a few pairs P_1 in the set $Bitangent(S_1)$ (step 1 of the algorithm). Indeed, assume that the probability for a pair P_1 in $Bitangent(S_1)$ to have no corresponding pair in $Bitangent(S_2)$ is q. After n iterations the probability that we didn't verify a right hypothesis is q^n. Then, if the verification stage is correct, the number of iterations will be very small.

We now focus on the complexity of each iteration. It is simply $|CloseDistance(P_1)| \times Verification\_Complexity$, where $Verification\_Complexity$ is the complexity of the verification stage. The number of pairs in $CloseDistance(P_1)$ depends on Δ. But even if Δ can be relatively small in practice, $|CloseDistance(P_1)|$ is porportional to $|Bitangent(S_2)|$, that is quasi-proportional to the number of points on S_2. Finally, $Verification\_Complexity$ is constant. Indeed, we can a priori fix the number n' of points in S_1' which are used to verify. Moreover, we can choose n' relatively small: the risk to accept a bad hypothesis is still very small. Indeed, assume that $(\mathbf{R}, \mathbf{t})$ is a rigid displacement which corresponds to a wrong hypothesis. If the points are randomly distributed within a volume and if for a point $M \in \mathbf{R}S_1 + \mathbf{t}$ the probability that there exists a point $N \in S_2$ such that $\|MN\| < \delta$ is p, then the probability to step the algorithm with this bad rigid displacement is

$$\sum_{i=k}^{i=n} \binom{i}{n} p^i (1 - p)^{n-i},$$

where $n = |S_1'|$ and $k = \rho \times n$. For instance, if $p = 0.4$, $\rho = 0.7$ and $n = 500$ this probability is approximatively 10^{-41}.

We tried to explain why the complexity of each iteration is quasi-linear in the number of points on the surfaces and why only a few iterations are necessary in practice. We do not pretend that this analysis is rigorous. It is not possible to set a value for p and q, and the points on the surfaces

are not randomly distributed (they belong to a smooth surface). But this can help to understand why, in practice, for our problems, the choice of the three parameters δ, Δ, and ρ is not difficult and why a good solution is found after a very small number of iterations (two or three).

APPENDIX C

Using the Generalized Mahalanobis Distance and χ^2 Tests to Deal with Outliers

Assume that $Match_j$ is perfect and that the computed geometric transformation perfectly superposes the two sets of points. For each pair (M_k, N_k) in $Match_j$, we can write the equation

$$\mathbf{f}_k(\mathbf{x}_k, \mathbf{a}) = \mathbf{0}, \tag{1}$$

where $\mathbf{x}_k$ is the vector obtained by concatenation of the M_k and N_k coordinates and $\mathbf{a}$ is the vector of the parameters of the geometric transformation.[17]

In reality, we cannot write Eq. (1) for all pairs (M_k, N_k) in $Match_j$. But we can assume that this is because each pair corresponds to a measurement $\hat{\mathbf{x}}_k$ of $\mathbf{x}_k$ corrupted with a random error $\mathbf{v}_k$:

$$\hat{\mathbf{x}}_k = \mathbf{x}_k + \mathbf{v}_k, A(\mathbf{v}_k) = \mathbf{0}, A(\mathbf{v}_k \mathbf{v}_k^t)$$
$$= \Lambda_k \geq 0, A(\mathbf{v}_i \mathbf{v}_j^t) = \mathbf{0} \; \forall i \neq j. \tag{2}$$

Given the measurements $\hat{\mathbf{x}}_k$, the EKF allows us to recursively compute an estimate $\hat{\mathbf{a}}_k$ of $\mathbf{a}$ and the associated covariance matrix $\mathbf{S}_k$ ($\hat{\mathbf{a}}_k = \mathbf{a} + \mathbf{w}_k$, where $\mathbf{w}_k$ is a random error: $A(\mathbf{w}_k) = \mathbf{0}$ and $A(\mathbf{w}_k \mathbf{w}_k^t) = \mathbf{S}_k \geq \mathbf{0}$). Only an initial estimate $(\hat{\mathbf{a}}_0, \mathbf{S}_0)$ is required and the minimized criterion is closely related (up to a first order approximation) to

$$C = (\hat{\mathbf{a}}_0 - \mathbf{a})^t \mathbf{S}_0^{-1} (\hat{\mathbf{a}}_0 - \mathbf{a})$$
$$+ \sum_{(M,N) \in Match_j} \mathbf{f}_k(\hat{\mathbf{x}}_k, \mathbf{a})^t \mathbf{W}_k^{-1} \mathbf{f}_k(\hat{\mathbf{x}}_k, \mathbf{a}), \quad (3)$$

where $\mathbf{W}_k = \widehat{\partial \mathbf{f}_k} / \partial \mathbf{x} \, \Lambda_k \, \widehat{\partial \mathbf{f}_k^t} / \partial \mathbf{x}$. Taking $\mathbf{S}_0^{-1} \approx \mathbf{0}$, C is the criterion we want to minimize. See Ayache [1] for more details.

Hence, in practice, the difficulty is just to choose the covariance matrix $\mathbf{S}_0$ associated with the initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$ of the rigid displacement and to choose the covariance matrix $\mathbf{W}_k$ associated with each point M_k. In practice,

we choose $\mathbf{W}_k$ corresponding to the covariance of the estimated error after the correct registration, and $\mathbf{S}_0$ as the diagonal matrix corresponding to the covariance of the error on the initial estimation of the rigid displacement parameters.

The problem of outliers is to distinguish between *likely* matches and *unlikely* matches in $Match_j$. For each pair (M_k, M_k), we consider that we have an estimate $\hat{\mathbf{a}}_{k-1}$ of $\mathbf{a}$ with associated covariance matrix $\mathbf{S}_{k-1}$ and a noisy measurement $(\hat{\mathbf{x}}_k, \Lambda_k)$ of $\mathbf{x}_i$, and we would like to then test the validity of the Eq. (1). If we consider the truncated expansion of $\mathbf{f}_k(\mathbf{x}_k, \mathbf{a})$

$$\mathbf{f}_k(\mathbf{x}_k, \mathbf{a}) = \mathbf{0} \approx \mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1}) + \frac{\widehat{\partial \mathbf{f}_k}}{\partial \mathbf{x}}(\mathbf{x}_k - \hat{\mathbf{x}}_k) + \frac{\widehat{\partial \mathbf{f}_k}}{\partial \mathbf{a}}(\mathbf{a}_k - \hat{\mathbf{a}}_k), \tag{4}$$

and assuming that $\mathbf{w}_k$ and $\mathbf{v}_k$ are zero-mean independent Gaussian noise, it is clear that $\mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1})$ is also—to within a first-order approximation—a Gaussian process whose mean and covariance are given by

$$A(\mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1})) = 0$$

$$\mathbf{Q}_i = A(\mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1}) \mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1})^t) = \frac{\widehat{\partial \mathbf{f}_k}}{\partial \mathbf{x}} \Lambda_k \frac{\widehat{\partial \mathbf{f}_k^t}}{\partial \mathbf{a}} \mathbf{S}_{k-1} \frac{\widehat{\partial \mathbf{f}_k^t}}{\partial \mathbf{a}}.$$

As a result, if the rank of $\mathbf{Q}_i$ is q, the generalized Mahalanobis distance

$$\delta(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1}) = [\mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1})]^t \mathbf{Q}_i^{-1} [\mathbf{f}_k(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1})] \tag{5}$$

is a random variable with a χ^2 probability distribution with q degrees of freedom.[18]

By consulting a table of values of the χ^2 distribution, it is easy to determine a confidence level ε for the Mahalanobis distance corresponding to, for example a 95% probability of having the distance (5) less than ε. In this case, we can consider as *likely* or *plausible* the measurements $\hat{\mathbf{x}}_k$ for which the inequality $\delta(\hat{\mathbf{x}}_k, \hat{\mathbf{a}}_{k-1}) < \varepsilon$ is verified, and consider any others as *unlikely* or *unplausible*.

APPENDIX D

Finding the Initial Estimate of the 3D–2D Curve Registration and of the 3D–2D Curve Registration

D.1. *The Algorithm for 3D–2D Curve Registration*

Let c be the 2D curve and C be the 3D curve. We first compute on c the set $Bitangent(c)$ of pair of points on c

---

[17] For example, for the 3D–3D rigid registration problem (Section 2.2), if $M_k = (x_1, y_1, z_1)$ and $N_k = (x_2, y_2, z_2)$ and if $(\alpha, \beta, \gamma, t_x, t_y, t_z)$ are the three Euler angles and translation parameters of the searched rigid displacement, we can have $\mathbf{x}_k = (x_1, y_1, z_1, x_2, y_2, z_2)$ and $\mathbf{a} = (\alpha, \beta, \gamma, t_x, t_y, t_z)$.

[18] If $q < p = $ the size of the measurement vector $\mathbf{f}_k$, $\mathbf{Q}_k^{-1}$ is the pseudo-inverse of $\mathbf{Q}_k$.

which share the same tangent lines. For each pair $pair_{2D}$ in $Bitangent(c)$ which corresponds to a tangent line t, we compute the set of points $Inter_{2D}(pair_{2D})$ which are intersections points between t and c. Then, we compute the set $CoplanarTangent(C)$ of pairs of points on C sharing coplanar tangent lines. For each $Pair_{3D}$ in $CoplanarTangent(C)$ with attached plane P, we compute the set of points $Inter_{3D}(Pair_{3D})$ which are intersection points between the plane P and the curve C. The algorithms for computing all these sets are given in Appendix A.

Let us define the two sets

$$Triplet_{2D} = \{(pair_{2D}, i)/pair_{2D} \in Bitangent(c) \\ and\ i \in Inter_{2D}(pair_{2D})\}$$

and

$$Triplet_{3D} = \{(Pair_{3D}, I)/Pair_{3D} \in CoplanarTangent(C)\ and \\ I \in Inter_{3D}(Pair_{3D})\}.$$

We can now describe the algorithm:

1. We choose a triplet (m_1, m_2, m_3) in $Triplet_{2D}$

2. For each triplet (M_1, M_2, M_3) in $Triplet_{3D}$, we compute the set $Hypotheses$ of the four projective transformations which map the 3D and the 2D triplets.

3. For each projective transformation in $Hypotheses$, we first verify that the projection of the tangent line attached to M_3 is the tangent line attached to m_3. If the test is positive, we apply the same verification scheme as described in Appendix B.1 for the rigid 3D–3D case: we compute the ratio $MatchedPoints$ of the number of points M on C such that the distance between the projection of M and its closest point on c is bellow a given distance δ, on the number of points on C. If $MatchedPoint$ is larger than $\rho(0 < \rho < 1)$ then we stop the algorithm and the result is the current projective transformation.

4. If no solution was found, we return to point 1.

D.2. *The Algorithm for 3D–2D Surface Registration*

To find the initial estimate $(\mathbf{R}_0, \mathbf{t}_0)$ of the rigid displacement such that $Proj\ o\ (\mathbf{R}_0, \mathbf{t}_0)$ is the projective transformation which transforms a surface S with the 2D occluding contour c, we first compute the set $Bitangent_{2D}(c)$ of pair of points which share the same tangent line on c and the set $Bitangent_{3D}(S)$ of pair of points which share the same tangent plane on S. We also compute a kd-tree $Tree$ based on the Euclidean coordinates of the normal vectors on the surface S which will allow us, given a normalized vector $\mathbf{m}$, to find quickly the set of normals $\mathbf{n}$ on S such that the angle between $\mathbf{m}$ and $\mathbf{n}$ is bellow a given threshold. Then we apply the following algorithm:

1. We choose a pair (m_1, m_2) in $Bitangent_{2D}(c)$ and a point m_3 on c.

2. For each pair (M_1, M_2) in $Bitangent_{3D}(S)$, we construct the set

$$Triplet_{3D}(M_1, M_2) = \{(M_1, M_2, M_3)/| \\ Angle(Normal_{2D}(m_1), Normal_{2D}(m_3)) - \\ Angle(Normal_{3D}(M_1), Normal_{3D}(M_3))| < \varepsilon\},$$

where $Angle$ is the function which returns the angle between two normalized vectors and ε is a given threshold. We use the kd-tree $Tree$ to compute efficiently this set.

3. For each triplet (M_1, M_2, M_3) in $Triplet_{3D}(M_1, M_2)$, we compute the set $Hypotheses$ of the four rigid displacement $(\mathbf{R}, \mathbf{t})$ such that $m_i = Proj(\mathbf{R}M_i + \mathbf{t})$.

4. For each rigid displacement $(\mathbf{R}, \mathbf{t})$ in $Hypotheses$, we check that the transformed surface verify the "fundamental property of occluding contour" (Subsection 4.1). To do this, we compute the ratio $MatchedPoint$ of the number of points on c which have a correspondent on $\mathbf{R}S + \mathbf{t}$, on the number of points on c. To determine if a point m has a correspondent on S, we proceed as follows. We first compute the set

$$CloseAngle(m) = \{M \in S/Angle(\mathbf{R}^tNormal_{2D}(m), \\ Normal_{3D}(M)) < \varepsilon\}.$$

Again, to compute efficiently the set $CloseAngle$, we use the kd-tree $Tree$. If there exists a point M in $CloseAngle(m)$ such that the 2D distance between m and $Proj(\mathbf{R}M + \mathbf{t})$ is bellow a given threshold δ, we consider that m has a correspondent. Otherwise, we consider that m has no correspondent for the rigid displacement $(\mathbf{R}, \mathbf{t})$. If the ratio $MatchedPoint$ is larger than a given threshold ρ ($0 < \rho < 1$), we stop the algorithm and the initial estimate is the projective transformation $Proj\ o\ (\mathbf{R}, \mathbf{t})$.

5. If no solution was found, we return to point 1.

D.3. *Complexity Analysis of the Initial Estimate Search for the 3D–2D Curve Registration Problem and for the 3D–2D Surface Registration Problem*

Basically, the complexity analysis for these two problems is not different from the analysis presented in Appendix B.2. Indeed the algorithms are very similar: instead of two set of pairs of points, we have two sets of triplets of points. The discussion for explaining why the number of global iterations is small, why the risk to stop the algorithm with a bad hypothesis is small, and why the verification stage can be performed in constant time would be very similar to the previous one. In fact, the complexity of the algorithms depends on the number of hypotheses to verify

at each iteration. This is the number N of 3D triplets which can correspond to a given 2D triplet.

For the 3D–2D curve registration, we have

$$N = |Triplet_{3D}| < |CoplanarTangent(C)|$$
$$\times \max_{Pair_{3D} \in CoplanarTangent(C)} |Inter_{3D}(Pair_{3D})|.$$

We explained in Appendix A.2 that $|CoplanarTangent(C)|$ is quasi-linear in the number of points in C. If C is not linear but smooth,

$$p = \max_{Pair_{3D} \in CoplanarTangent(C)} |Inter_{3D}(Pair_{3D})|$$

is small.[19] Finally, the complexity of the algorithm is $O(p \times n)$, where p is a small integer and n is the number of points describing C.

For the 3D–2D surface registration, we have

$$N = |Bitangent_{3D}(S)| \times \max_{(M_1, M_2) \in Bitangent_{3D}(S)} |Angle_\varepsilon(M_1)|,$$

where

$$Angle_\varepsilon(M_1) = \{M_3 \in S/$$
$$Angle(Normal_{3D}(M_1), Normal_{3D}(M_3)) -$$
$$Angle(Normal_{2D}(m_1), Normal_{2D}(m_3))| \leq \varepsilon\}$$

(ε is a given threshold).

We explained in Appendix A.1 that $|Bitangent_{3D}(S)|$ is quasi-linear in the number of points in S. $p = \max_{(M_1, M_2) \in Bitangent_{3D}(S)} |Angle_\varepsilon(M_1)|$ is more difficult to estimate. In practice, assuming that the normals on the surface S are approximately uniformly distributed on the unit sphere, we try to choose m_3 such that $\beta = \pi$.[20] Hence, p is small. Finally, the complexity is roughly $O(p \times n)$, where n is the number of points on S. Because p is small, the algorithm is efficient in practice.

APPENDIX E

Decomposition of the Calibration Matrix T

We explain in this appendix why it is possible to assume that the camera image formation corresponds to a perspective projection of center $O = (0, 0, 0)$ on a focal plane $FocalPlane$ of equation $z = 1$ without loss of generality.

The result of the calibration process is a 3×4 matrix $\mathbf{T}$. This is the matrix of the 3D–2D projective transformation which maps a reference object of known shape (the calibration grid) onto an image of this object. This matrix $\mathbf{T}$ can be decomposed into $\mathbf{T} = \mathbf{AD}$. $\mathbf{D}$ is the 3×4 matrix of the *extrinsic parameters* and it is a matrix of change of frame from the calibration grid frame to the camera frame. $\mathbf{A}$ is a 3×3 matrix such that

$$\mathbf{A} = \begin{pmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ & 0 & 1 \end{pmatrix} \tag{6}$$

It is called the matrix of the *intrinsic parameters* and describe the camera image formation process. Basically, it is an affine transformation of *FocalPlane*.

Let us recall why this decomposition is possible (the proof is constructive). We note $\mathbf{T} = [\mathbf{H}|\mathbf{t}]$ ($\mathbf{H}$ is a 3×3 matrix and $\mathbf{t}$ a 3×1 matrix). For any matrix $\mathbf{H}$, $\mathbf{HH}^T$ is clearly a symmetric matrix. If, moreover, $\mathbf{H}$ is a square nonsingular matrix, then $\mathbf{HH}^T$ is also positive definite. Indeed, for each vector $\mathbf{v}$, different from $\mathbf{0}$,

$$\mathbf{v}^T\mathbf{HH}^T\mathbf{v} = \|\mathbf{H}^T\mathbf{v}\|^2 > 0,$$

since the kernel of $\mathbf{H}^T$ reduced to $\mathbf{0}$ because of the nonsingularity of $\mathbf{H}$. Consequently, a nonsingular upper triangular matrix $\mathbf{U}$ exists such that

$$\mathbf{HH}^T = \mathbf{UU}^T$$

(it is one form of the Cholesky decomposition). If we denote by $\mathbf{O}$ the matrix $\mathbf{U}^{-1}\mathbf{H}$, we see that

$$\mathbf{OO}^T = \mathbf{U}^{-1}\mathbf{HH}^T\mathbf{U}^{-1T} = \mathbf{I}$$

which shows that $\mathbf{O}$ is an orthogonal matrix. Hence, we have the following decomposition of $\mathbf{H}$: $\mathbf{H} = \mathbf{UO}$, where $\mathbf{U}$ is a nonsingular upper triangular matrix and $\mathbf{O}$, an orthogonal matrix. Finally, $\mathbf{A}$ is the result of the division $\mathbf{U}/u_{33}$ which is possible because these matrices are defined up to a multiplication factor. $\mathbf{D}$ is simply $\mathbf{D} = [\mathbf{O}|\mathbf{A}^{-1}\mathbf{t}]$.

In practice, before starting the registration process, we first decompose $\mathbf{T}$ as explained upper. Then, we replace each point (x, y) on the 2D curve by (x', y'), such that

$$(x', y', 1)^t = \mathbf{A}^{-1}(x, y, 1)^t.$$

Working with the new set of 2D points, because $\mathbf{A}$ is an affine transformation of *FocalPlane,* everything happens as if the camera image formation process was a perspective

---

[19] This is the same kind of assumption as in Appendix A.1.
[20] Most often in our applications, the contours are closed and the distance from the object to the camera is large with respect to the size of the object, therefore it is possible to choose such a point.

projection of center $O = (0, 0, 0)$ on a focal plane *FocalPlane* of equation $z = 1$.

REFERENCES

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

2. N. Ayache, Analysis of three-dimensional medical images—Results and challenges, Research Report 2050, INRIA, 1993.

3. R. Basri and S. Ullman, The alignment of objects with smooth surfaces, *Comput. Vision Graphics Image Process.* **57**(3), 1993, 331–345.

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

5. G. Champleboux, S. Lavallée, R. Szeliski, and L. Brunie, From accurate range imaging sensor calibration to accurate model-based 3-D object localization, in *Proceedings of the IEEE Conference on Vision and Pattern Recognition, Urbana Champaign, June 1992.*

6. Y. Chen and G. Medioni, Object modeling by registration of multiple range images, *Image Vision Comput.* **10**(3), 1992, 145–155.

7. P. Cinquin, P. Demongeot, J. Troccaz, S. Lavallee, G. Champleboux, L. Brunie, F. Leitner, P. Sautot, B. Mazier, A. Perez M. Djaid, T. Fortin, M. Chenin, and A. Chapel, Igor: Image guided operating robot. Methodology, medical applications, results, *ITBM* **13** 1992, 373–393.

8. L. Cohen, Use of auxiliary variables in computer vision problems, *Cahiers de mathematiques de la décision,* (9409), Feb. 1994.

9. A. C. F. Colchester, J. Zhao, C. Henri, R. L. Evans, P. Roberts, N. Maitland, D. J. Hawkes, D. L. G. Hill, A. J. Strong, D. G. Thomas, M. J. Gleeson, and T. C. S. Cox, Craniotomy simulation and guidance using a stereo video based tracking system (vislan), in *Visualization in Biomedical Computing, Rochester, MN, Ot 1994.*

10. P. E. Danielsson, Euclidean distance mapping, *Comput. Graphics Image Process.* **14** 1980, 227–248.

11. F. Devernay and O. D. Faugeras, Computing differential properties of 3d shapes from stereoscopic images without 3d models, in *IEEE Proceedings of Computer Vision and Pattern Recognition 1994 (CVPR'94), Seattle, June 1994.*

12. O. Faugeras and M. Hébert, The representation, recognition and locating of 3*d* objects, *Int. J. Robotics Res.* **5**(3), 1986, 27–52.

13. J. Feldmar and N. Ayache, Rigid, affine and locally affine registration of free-form surfaces, Research Report 2220, INRIA, March 1994. Accepted for publication in *Int. J. Comput. Vision.* Published in two parts in ECCV'94 (rigid and affine) and CVPR'94 (locally affine).

14. P. Fua and Y. G. Leclerc, Registration without correspondences, in *IEEE Proceedings of Computer Vision and Pattern Recognition 1994 (CVPR'94), Seattle, June 1994.*

15. L. Van Gool, T. Moons, E. Pauwels, and A. Oosterlinck, Semi-differential invariants, in *Geometric Invariance in Computer Vision* (J. L. Mundy and A. Zisserman, Eds.), MIT Press, Cambridge, MA, 1992.

16. W. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints,* MIT Press, Cambridge, MA, 1990.

17. W. E. L. Grimson, T. Lozano-Perez, W. M. Wells III, G. J. Ettinger, S. J. White, and R. Kikinis, An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization, in *IEEE Proceedings of Computer Vision and Pattern Recognition 1994 (CVPR '94), Seattle, June 1994.*

18. A. Guéziec and N. Ayache, Smoothing and matching of 3-D-space curves, *Int. J. Comput. Vision* **12**(1), 1994.

19. R. M. Haralick and Yu Hong Chu, Solving camera parameters from the perspective projection of a parameterized curve, *Pattern Recognit.* **17**(6), 1984, 637–645.

20. T. Joshi and J. Ponce, Hot curves for modelling and recognition of smooth curved 3d objects, in *IEEE Proceedings of Computer Vision and Pattern Recognition 1994 (CVPR '94), Seattle, June 1994.*

21. S. Lavallée, R. Szeliski, and L. Brunie, Matching 3-d smooth surfaces with their 2-d projections using 3-d distance maps, in *SPIE, Geometric Methods in Computer Vision, San Diego, CA, July 1991.*

22. L. Lemieux, R. Jagoe, D. R. Fish, N. D. Kitchen, and D. G. Thomas, A patient-to-computed-tomography image registration method based on digitally reconstructed radiographs, *Med. Phys.* **21,** Nov. 1994.

23. A. Linnainmaa, D. Harwood, and L. S. Davis, Pose determination of a three-dimensional object using triangle pairs, *IEEE Trans. Pattern Anal. Machine Intell.* **10**(5), 1988, 635.

24. G. Malandain, *Filtrage, topologie et mise en correspondance d'images médicales multidimensionnelles,* Ph.D. thesis, Ecole Centrale de Paris, Sept. 1992.

25. G. Malandain, G. Bertrand, and N. Ayache, Topological segmentation of discrete surfaces, *IJCV* **10**(2), 1993, 183–197.

26. H.-T. Yau, C.-H. Menq, and G.-Y. Lai, Automated precision measurement of surface profile in cad-directed inspection, *IEEE Trans. RA* **8**(2), 1992, 268–278.

27. X. Pennec and J. P. Thirion, Validation of registration methods based on points and frames, research report, INRIA, Dec. 1994. Submitted to ICCV'95.

28. A. Shweikard, R. Tombropoulos, J. Adler, and J. C. Latombe, Planning for image-guided radiosurgery, in *AAAI 1994 Spring Symposium Series. Application of Computer Vision in Medical Image Processing, Stanford University, March 1994.* Also in *Proc. IEEE Int. Conf. Robotics and Automation, 1994.*

29. D. Simon, M. Hebert, and T. Kanade, Techniques for fast and accurate intra-surgical registration, in *First International Symposium on Medical Robotics and Computer Assisted Surgery, Pittsburgh, Sept. 1994.*

30. Richard Szeliski and Stéphane Lavallée, Matching 3-d anatomical surfaces with non-rigid volumetric deformations, in *Proceedings of the IEEE Workshop on Biomedical Images Analysis* (*WBIA '94*), *Seattle, June 1994.* Also in *AAAI 1994 Spring Symposium Series. Application of Computer Vision in Medical Image Processing, Stanford University, 1994.*

31. R. Taylor, An overview of computer assisted surgery research, *International Symposium on Robotic Research, Oct. 1993.*

32. R. H. Taylor, S. Lavallee, G. C. Burdea, and R. W. Mosge, *Computer Integrated Surgery,* MIT Press, Cambridge, MA, to appear.

33. J. P. Thirion and A. Gourdon, The 3-D marching lines algorithm and its application to crest lines extraction, Research Report 1672, INRIA, May 1992.

34. P. van den Elsen, E. J. D. Pol, and M. Viergever, Medical image matching: A review with classification, *IEEE Eng. Med. Biol.* **12**(4), 1993, 26–39.

35. Paul Viola and William M. Wells III, Alignment by maximisation of mutual information, in *Proceedings of the Fifth International Conference on Computer Vision* (*ICCV '95*), *Boston, June 1995.*

36. Z. Zhang, Iterative point matching for registration of free-form curves and surfaces, *Int. J. Comput. Vision* **13**(2), 1994, 119–152. Also Research Report No. 1658, INRIA Sophia-Antipolis, 1992.