From voxel to curvature

Olivier Monga, Nicholas Ayache, Peter T. Sander

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

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

Rapports de Recherche

N° 1356

Programme 6
Robotique, Image et Vision

FROM VOXEL TO CURVATURE

Olivier MONGA, Nicholas AYACHE
Peter T. SANDER

Décembre 1990

RR-1356

Programme 4

Robotique, Image et Vision

From voxel to curvature [1]

Olivier Monga
Nicholas Ayache
INRIA Domaine de Voluceau-Rocquencourt - B.P. 105
78153 Le Chesnay Cedex, France

Peter T. Sander
INRIA, Unité de Recherche Sophia-Antipolis
06565 Valbonne Cedex, France.

[1]This work was partially supported by Digital Equipment Corporation and European AIM (Advanced Informatics in Medicine) Project Murim.

Abstract

We establish a theoretical link between the 3D edge detection and the local surface approximation using uncertainty. As practical application of the theory, we present a method for computing typical curvature features from 3D medical images. We determine the uncertainties inherent in edge (and surface) detection in 2- and 3-dimensional images by quantitatively analyzing the uncertainty in edge position, orientation and magnitude produced by the multidimensional (2-D and 3-D) versions of the Monga-Deriche-Canny recursive separable edge-detector. The uncertainty is shown to depend on edge orientation, e.g., the position uncertainty may vary with a ratio larger than 2.8 in the 2-D case, and 3.5 in the 3-D case. These uncertainties are then used to compute local geometric models (quadric surface patches) of the surface, which are suitable for reliably estimating local surface characteristics, for example, Gaussian and mean curvature. We demonstrate the effectiveness of our methods compared to previous techniques. These curvatures are then used to obtain more structured features such as curvature extrema and lines of curvature extrema. The final goal is to extract robust geometric features on which registration and/or tracking procedures can rely.

Des voxels aux courbures

Résumé

Nous établissons un lien théorique entre la détection de contours 3D et la modélisation des surfaces en utilisant les incertitudes. Comme application pratique, nous présentons une méthode pour calculer des attributs caractéristiques liés aux courbures a' partir d'images médicales 3D. Nous déterminons les incertitudes inhérentes à la detection de contours dans des images 2D et 3D en analysant quantitativement les incertitudes sur la position des contours, l'orientation et la norme du gradient dans le cas de l'opérateur Deriche. Nous montrons que ces incertitudes dépendent de l'orientation du contour, par exemple l'incertitude sur la position varie avec un facteur 2.8 dans le cas 2D et 3.5 dans le cas 3D. Nous utilisons ensuite ces incertitudes pour le calcul de modèles géométriques locaux de la surface qui permettent l'estimation d'attributs caractéristiques, par exemple les courbures Gaussiennes et moyennes. Nous montrons l'éfficacité de nos algorithmes par rapport aux méthodes antérieures. Ces courbures permettent ensuite d'obtenir des attributs de plus haut niveau tels que les extrema de courbure et les lignes d'extrema de courbure. Le but final est d'extraire des attributs géométriques robustes sur lesquels les procédures de fusion de données et de mise en correspondance peuvent s'appuyer.

Introduction

Modern medical imaging techniques, such as Nuclear magnetic Resonance or X-ray computed tomography provide three dimensional (3D) images of internal structures of the body, usually by means of a stack of tomographic images. In many applications, the physician asks for a segmentation of these 3D images into regions of interest he wants to manipulate, display, and characterize by objective measurements [AB+90]. The first stage in the automatic analysis of such data is 3D edge detection [ZH81, MD89, MDR91] which provide points corresponding to the boundaries of the surfaces forming the 3D structure. The next stage is to characterize the local geometry of these surfaces in order to extract points or lines on which registration and/or tracking procedures can rely.

Sander and Zucker have proposed computing surface singularities by the calculation of curvatures using local approximation and by iterative refinement of the curvature field [SZ87, SZ90]. In this paper we present a pipeline of process which define a hierarchical description of the second order differential characteristics of the surfaces. We focus on the theoretical coherence of these levels of representation. Our levels of representation of the local geometry of the surfaces are :

- 3D edge points,
- Mean and Gaussian curvature, Principal curvature directions,
- Local images of the curvatures,
- Smoothed curvature field,
- Characteristic points : curvature extrema, parabolic points, umbilic points ...
- Characteristic lines : extrema curvature line, parabolic lines ...

Three-dimensional edge detection is performed using recursive separable filters approximating the gradient or Laplacian as described in [MD89, MDR91, MDMC90]. From these edge points we build an adjacency graph with position and gradient vector attached to each edge point.

To compute curvatures from this graph we fit a local model at the neighbourhood of each point. The local model is a quadratic surface and the fitting method is a Kalman filter. Our approximation scheme uses the locations of the edge points and also the gradient direction which approximates the normal to the surface. Using uncertainty we establish a theoretical link between the edge detection and the local surface approximation by addressing the following two issues :

- How to determine the uncertainties inherent in edge detection in 2D or 3D images,
- How to incorporate these uncertainties into the computation of local geometric models.

In particular, we calculate the uncertainty of edge location, direction and magnitude for the 3D Deriche operator. Our statistical results are then used as a solid theoretical foundation on which to base subsequent computations, such as, for example, the determination of local surface curvature using local geometric models or for surface segmentation [BJ88].

We tie the results of the analysis of the uncertainties involved in edge detection to the estimation of local geometric surfaces by a Kalman filtering technique. While the instantiation of these local models is similar to the method of Sander and Zucker [SZ90], the utilization of uncertainties yields improved results. In addition, Kalman filtering permits incremental and selective incorporation of new data, thus ensuring that the local models are fit to, and only to, relevant data points. We expect that this will permit us to effectively deal with the problem of discontinuities where the local surface smoothness assumptions break down.

From the local fitting, we calculate for each edge point a mean curvature, a gaussian curvature, and principal curvature directions, and covariance matrices defining the uncertainty. We define local curvature images by projecting at each point the curvatures of the neighbours onto the tangent plane. This yields the local aspect of the curvatures at a point. These local images of the curvatures are used to improve the continuity of the mean and Gaussian curvature fields. This could be done by iterative linear refinement as proposed in [SZ90]. We propose using a continuity model derived from our local geometry model (a quadratic) and a Least Median Squares Estimator to instantiate it [PM90, JJ90]. The first choice allows coherence with the local geometric model, the second one ensures that results will not be spoiled by outlyers.

From the smoothed curvature field we extract typical points such as curvature extrema, parabolic points, umbilic points ... This is done by using the local image of the curvatures defined at each (edge) point. For instance to select the maxima of the mean curvature in the direction of its gradient we employ a two-stage algorithm. The first stage consists in approximating the gradient of the mean curvature at a point by using a linear gradient operator. The second one is very similar to the extraction of the extrema of the gradient magnitude in the gradient direction used in many 2D edge detection algorithms [Der87, Can86].

From these typical points, we extract characteristic lines such as line of mean curvature extrema, parabolic lines .... To obtain lines of mean curvature extrema, we perform a 3D hysteresis thresholding on the extrema curvature points using the mean curvature. Here again the algorithm is very similar to the one used to threshold the local gradient extrema described in [MDR91].

The paper is organized as follow. In the first section we show how uncertainty allows unifying the 3D edge detection and the surface fitting. We deal first with the general problem of uncertainty in surface fitting, and then with the actual determination of the uncertainty from our particular operator (Deriche operator). In subsection 1 we introduce the local geometric models and curvature computations. Subsection 2 sets up the local surface fitting problem and the Kalman filter formalism for coping with uncertain data. Subsection 3 determines the inherent uncertainty resulting from the 3-D gradient operator, and how this is related to the Kalman filter.

In the second section we deal with the use of these curvatures to extract typical curvature features. We utilize the projection on the tangent plane of local images of curvatures which is frame invariant. Thus we reduce the extraction of typical curvature features to the filtering of the local images of curvatures.

The third section presents some experimental results for 3D synthetic images and real data. For real data we demonstrate the stability of the mean and Gaussian curvatures using two 3D scanner images of the same organ taken at different positions.

1 Link between 3D edge detection and surface fitting

1.1 Problem formulation

In this section, we set up the local parametric surface models, describe briefly how to compute surface curvatures from the models, and present the problem of determining the parameters of the models from the data derived from the 3-D gradient operator. We thus assume that we are given the locations of estimated surface points P and their estimated surface normals $\mathbf{N}$ corresponding to the vector gradient, (see [MD89, MDR91] on how to compute them). Using $\mathbf{N}$ we can establish a *tangent plane coordinate system* at P, which we denote ($\mathbf{P}$, $\mathbf{Q}$, $\mathbf{N}$). Note that the basis ($\mathbf{P}$, $\mathbf{Q}$) of the tangent plane at P is arbitrary — the only constraint is that the coordinate system be right-handed and orthonormal.

In the following, P is the point at which the surface patch is being fit, and Q_i are neighbouring estimated surface points with associated normals $\mathbf{n}_i$, with both given in P's tangent plane coordinates (the development is simpler in these coordinates; we map everything into the actual image coordinates in §1.2.3).

1.1.1 Local geometric model

We assume that the data returned by the gradient operator represent noisy estimates of points and normals from a (smooth) surface $\mathcal{S}$. We treat a surface as a differentiable manifold and build local charts (parametrizations) at all the estimated surface points. Thus, at $P \in \mathcal{S}$

we assume that the local chart (ψ, U)

$$\psi : U \subset \mathcal{S} \to \mathbb{R}^2$$

(ψ a diffeomorphism, $P \in$ open set U) is such that $\psi(P) = (0,0)$ and its imbedding

$$\phi = i \circ \psi^{-1} : \psi(U) \subset \mathbb{R}^2 \to \mathbb{R}^3$$

in $\mathbb{R}^3$ (based on P's tangent plane coordinates) is the graph of some function $h : \psi(U) \to \mathbb{R}$ with

$$\begin{aligned} h(0,0) &= 0, \\ h_p(0,0) = \left.\frac{\partial h}{\partial p}\right|_{(0,0)} &= 0 \\ h_q(0,0) = \left.\frac{\partial h}{\partial q}\right|_{(0,0)} &= 0 \end{aligned}$$

(this is always true in some local chart). The Taylor expansion of h about the origin is

$$h(p,q) = \frac{1}{2}\left(h_{pp}p^2 + 2h_{pq}pq + h_{qq}q^2\right) + R.$$

Since our ultimate goal is computation of curvatures and related information, we take the simplest local chart which is appropriate, i.e., where

$$h = \frac{1}{2}ep^2 + fpq + \frac{1}{2}gq^2, \tag{1}$$

where we write

$$e = h_{pp}(0,0), \quad f = h_{pq}(0,0), \quad g = h_{qq}(0,0)$$

(known as a *parabolic quadric*).

1.1.2 Surface curvature

The curvature of the surface $\mathcal{S}$ at P can be computed from its local parametrization $\phi : \psi(U) \to \mathbb{R}^3$ in P's tangent plane coordinates. The surface normal at P is expressed as

$$\mathbf{N}(0,0) = \left.\frac{\phi_p \times \phi_q}{\|\phi_p \times \phi_q\|}\right|_{(0,0)}.$$

(In the following, we take it as understood that derivatives are evaluated at $(0,0)$, i.e., the $|_{(0,0)}$ is implicit.) The matrices

$$\begin{aligned} F_1 &= \begin{pmatrix} \langle \phi_p, \phi_p \rangle & \langle \phi_p, \phi_q \rangle \\ \langle \phi_q, \phi_p \rangle & \langle \phi_q, \phi_q \rangle \end{pmatrix}, \\ F_2 &= \begin{pmatrix} -\langle \phi_p, \mathbf{N}_p \rangle & -\langle \phi_p, \mathbf{N}_q \rangle \\ -\langle \phi_q, \mathbf{N}_p \rangle & -\langle \phi_q, \mathbf{N}_q \rangle \end{pmatrix}, \end{aligned}$$

are determined from the first and second fundamental forms respectively of the surface ($(\bullet,\bullet)$ denotes inner product). The principal curvatures κ_1, κ_2 of ϕ at P in (**P**, **Q**, **N**) coordinates are the two eigenvalues of the matrix $F_2 F_1^{-1}$, and the Gaussian and mean curvatures are

$$\begin{aligned} \kappa_g &= \kappa_1 \kappa_2, \\ \kappa_m &= \frac{\kappa_1 + \kappa_2}{2} \end{aligned}$$

respectively. We show how to compute uncertainties in the curvatures in §1.2.3.

1.2 Instantiating the model

Now, we wish to determine the local quadric surface passing through point P which "best" (in a sense made precise below) fits neighbouring points $Q_i = (p_i, q_i, n_i)^t$ and their normals $\mathbf{n}_i = (\alpha_i', \beta_i', \gamma_i')^t$. In P's tangent plane coordinate system, the equation of the quadric gives us a first measurement equation

$$E_1(e, f, g) = p_i^2 e + 2 p_i q_i f + q_i^2 g - 2 n_i = 0 \tag{2}$$

between the position of Q_i and the parameters e, f, g which are to be determined from the data.

In addition to estimated surface point locations, the 3-D gradient operator provides an estimate of gradient direction and we can use the measured normal $\mathbf{n}_i$ at point Q_i to further constrain the quadric surface parameters. We know that, in the tangent plane coordinates, the quadric's normal at point Q_i is

$$\mathbf{n}_i = \begin{pmatrix} -p_i e - q_i f \\ -p_i f - q_i g \\ 1 \end{pmatrix}.$$

Denoting the scaled normal *measured* at point Q_i by $(\alpha_i, \beta_i, 1)^t = (\alpha_i'/\gamma_i', \beta_i'\gamma_i', \gamma_i'/\gamma_i')^t$, we obtain two more measurement equations

$$E_2(e, f, g) = p_i e + q_i f + \alpha_i = 0, \tag{3}$$

$$E_3(e, f, g) = p_i f + q_i g + \beta_i = 0. \tag{4}$$

Equations (E_1–E_3) are the three measurement equations which constrain the determination of the parameters e, f, g of the quadric at point P. These equations should be compared to the four equations E_1'–E_4' of [SZ90]: Eq. E_1 is the same, but Eqs. (E_2' - E_4') there were based on unit normals and involved a non-linear combination of e, f, g. The only restriction on our equations here is the assumption that the normal $\mathbf{n}_i$ of Q_i measured in the tangent

plane coordinates of P has a nonzero third component, which is reasonable if we assume that Q_i lies in the neighborhood of point P. (In fact, if the surface is regular, such a neighborhood exists [dC76, p.164], at least before discretization[1]). When this component vanishes, the local parametrization of the quadric in these coordinates is no longer valid, and point Q_i should not be taken into account.

Denoting

$$\mathbf{A}_i = \begin{pmatrix} p_i^2 & 2p_iq_i & q_i^2 \\ p_i & q_i & 0 \\ 0 & p_i & q_i \end{pmatrix}, \qquad \mathbf{b}_i = \begin{pmatrix} 2n_i \\ -\alpha_i \\ -\beta_i \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} e \\ f \\ g \end{pmatrix},$$

the measurement Eqs. (E_1-E_3) at P can be put in matrix form

$$\mathbf{A}_i\mathbf{x} = \mathbf{b}_i.$$

1.2.1 Non-recursive minimum variance least-squares solution

We wish to weight the measurement equations by the uncertainty of our measured parameters, i.e., the coordinates of points Q_i and attached normals $\mathbf{n}_i$. Once this is done (cf. next sections), we end up with a matrix $\mathbf{W}_i$ which is the covariance of $\mathbf{A}_i\mathbf{x} - \mathbf{b}_i$

$$\mathbf{W}_i = E\left[(\mathbf{A}_i\mathbf{x} - \mathbf{b}_i)(\mathbf{A}_i\mathbf{x} - \mathbf{b}_i)^t\right].$$

Then a weighted least-squares solution $\mathbf{x}$ to our problem at P using all $Q_i, i = 1, \ldots, n$ in some neighbourhood will therefore minimize

$$C = \sum_i (\mathbf{A}_i\mathbf{x} - \mathbf{b}_i)^t \mathbf{W}_i^{-1} (\mathbf{A}_i\mathbf{x} - \mathbf{b}_i),$$

and is given by

$$\mathbf{x} = \left(\mathbf{A}^t\mathbf{W}^{-1}\mathbf{A}\right)^{-1} \mathbf{A}^t\mathbf{W}^{-1}\mathbf{b},$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_n \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_n \end{pmatrix}, \qquad \text{and} \quad \mathbf{W} = \begin{pmatrix} \mathbf{W}_1 & & \\ & \ddots & \\ & & \mathbf{W}_n \end{pmatrix}.$$

1.2.2 Recursive solution

In fact, we implement a recursive solution to this problem, better known as a *Kalman filter* [Lue69, HZ83, Aya91]. By this method, each time a new measurement Q_i is given, it

[1] A common enough assumption throughout computer vision.

is only necessary to compute $\mathbf{A}_i$ and $\mathbf{b}_i$ *for that point* and to update the current solution $(\mathbf{x}_i, \mathbf{S}_i)$ using the recursive equations

$$\begin{cases} \mathbf{x}_i &= \mathbf{x}_{i-1} + \mathbf{K}_i(\mathbf{b}_i - \mathbf{A}_i\mathbf{x}_{i-1}), \\ \mathbf{K}_i &= \mathbf{S}_{i-1}\mathbf{A}_i^t(\mathbf{W}_i + \mathbf{A}_i\mathbf{S}_{i-1}\mathbf{A}_i^t)^{-1}, \\ \mathbf{S}_i &= (\mathbf{I} - \mathbf{K}_i\mathbf{A}_i)\mathbf{S}_{i-1}. \end{cases}$$

$\mathbf{S}_i$ is the *parameter* covariance matrix

$$\mathbf{S}_i = E\left[(\mathbf{x}_i - \mathbf{x})(\mathbf{x}_i - \mathbf{x})^t\right]$$

relating the current estimate $\mathbf{x}_i$ and the ideal value $\mathbf{x}$ of the parameter vector. This is a measure of the quality of our estimate — a small covariance means that the computed estimate $\mathbf{x}_i$ is expected to lie close to the "actual" parameters $\mathbf{x}$. It is necessary to initialize the filter with $(\mathbf{x}_0, \mathbf{S}_0)$, which can be taken as $\mathbf{x}_0 = \mathbf{0}, \mathbf{S}_0 = \infty\,\mathbf{I}$ when *no a priori* information is available about any of the parameters.

1.2.3 Practical details

For simplicity, the preceding development was presented with all data assumed to be in P's tangent plane coordinate system. We now show how to transform from the actual data points Q_i and normal vectors $\mathbf{n}_i$, each with their respective associated covariance matrices $\mathbf{W}_Q$ and $\mathbf{W}_\mathbf{n}$, measured in a *global* coordinate system, i.e., the coordinate system of the image, to variables $\mathbf{v} = (p_i, q_i, n_i, \alpha_i, \beta_i)^t$ and associated covariance matrix $\mathbf{W}_i$ in P's tangent plane coordinates. Thus we can apply the above theory directly to the image data.

Computing parameters v

We now assume that $Q_i = (x_i, y_i, z_i)^t$ and $\mathbf{n}_i = (n_{x_i}, n_{y_i}, n_{z_i})^t$ are given in a global coordinate system $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. To express them in P's tangent plane coordinates $(\mathbf{P}, \mathbf{Q}, \mathbf{N})$ at point $P = (x, y, z)^t$, we compute

$$\begin{pmatrix} p_i \\ q_i \\ n_i \end{pmatrix} = \mathbf{R}\begin{pmatrix} x_i - x \\ y_i - y \\ z_i - z \end{pmatrix}, \qquad \begin{pmatrix} \alpha_i' \\ \beta_i' \\ \gamma_i' \end{pmatrix} = \mathbf{R}\begin{pmatrix} n_{x_i} \\ n_{y_i} \\ n_{z_i} \end{pmatrix},$$

and

$$\alpha_i = \frac{\alpha_i'}{\gamma_i'}, \qquad \beta_i = \frac{\beta_i'}{\gamma_i'},$$

for

$$\mathbf{R} = \begin{pmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \\ N_x & N_y & N_z \end{pmatrix}.$$

The coordinates of the tangent plane basis vectors at P expressed in the global image coordinates $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ are

$$\mathbf{P} = \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix}, \qquad \mathbf{Q} = \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix}, \qquad \mathbf{N} = \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix}.$$

Computing covariances $\mathbf{W}_i$

We assume that the covariance of point Q_i and its normal $\mathbf{n}_i$ are given in the *global coordinate system* $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ by $\mathbf{W}'_{Q_i}$ and $\mathbf{W}'_{\mathbf{n}_i}$ respectively. Since, for any affine transformation of a random variable $\mathbf{v}_i \rightarrow \mathbf{w}_i = \mathbf{M}(\mathbf{v}_i - \mathbf{v})$ we have

$$E\left[(\mathbf{w}_i - \mathbf{w})(\mathbf{w}_i - \mathbf{w})^t\right] = \mathbf{M}\, E\left[(\mathbf{v}_i - \mathbf{v})(\mathbf{v}_i - \mathbf{v})^t\right] \mathbf{M}^t,$$

the corresponding covariance matrices expressed in P's tangent plane coordinate system $(\mathbf{P}, \mathbf{Q}, \mathbf{N})$ are

$$\mathbf{W}_{Q_i} = \mathbf{R}\mathbf{W}'_{Q_i}\mathbf{R}^t, \qquad \mathbf{W}_{\mathbf{n}_i} = \mathbf{R}\mathbf{W}'_{\mathbf{n}_i}\mathbf{R}^t,$$

which are 3×3 matrices.

In fact, if we express $\mathbf{n}_i = (\alpha'_i, \beta'_i, \gamma'_i)^t$ in the tangent plane coordinates, we must compute the covariance of $\alpha_i = \alpha'_i/\gamma'_i$ and $\beta_i = \beta'_i/\gamma'_i$. As a first order approximation, we compute the 2×2 matrix $\widetilde{\mathbf{W}}_{\mathbf{n}_i}$

$$\widetilde{\mathbf{W}}_{\mathbf{n}_i} = \mathbf{J}_1 \mathbf{W}_{\mathbf{n}_i} \mathbf{J}_1^t$$

where $\mathbf{J}_1$ is the Jacobian matrix

$$\mathbf{J}_1 = \begin{pmatrix} \frac{1}{\gamma'_i} & 0 & \frac{-\alpha'_i}{\gamma'^2_i} \\ 0 & \frac{1}{\gamma'_i} & \frac{-\beta'_i}{\gamma'^2_i} \end{pmatrix}$$

of the change of variables. Therefore the 5×5 matrix

$$\mathbf{W}_{Q,\mathbf{n}} = \begin{pmatrix} \mathbf{W}_Q & \mathbf{0}_{3\times 2} \\ \mathbf{0}_{2\times 3} & \widetilde{\mathbf{W}}_{\mathbf{n}_i} \end{pmatrix}$$

is the covariance of our measurement vector $(p_i, q_i, n_i, \alpha_i, \beta_i)$.

The 3×3 covariance matrix $\mathbf{W}_i$ is computed as a first order approximation by

$$\mathbf{W}_i = \mathbf{J}_2 \mathbf{W}_{Q,\mathbf{n}_i} \mathbf{J}_2^t,$$

where $\mathbf{J}_2$ is the Jacobian matrix

$$\mathbf{J}_2 = \begin{pmatrix} 2p_i e + 2q_i f & 2p_i f + 2q_i g & -2 & 0 & 0 \\ e & f & 0 & 1 & 0 \\ 0 & f & g & 0 & 1 \end{pmatrix}.$$

The 2×2 covariance matrix of curvatures is determined similarly,

$$\mathbf{W}_c = \mathbf{J}_3 \mathbf{W}_{Q_i} \mathbf{J}_3^t,$$

with the Jacobian

$$\mathbf{J}_3 = \begin{pmatrix} c & -2b & a \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}.$$

1.3 Estimating error in edge detection

In this section, we determine the uncertainty inherent in edge detection in digital images by considering the 3-D (modified Canny) edge detector of Monga and Deriche [MD89, MDR91]. We first deal with uncertainty in edge position, and then with the error in edge direction and magnitude. We determine precisely the covariance matrices needed for the local quadric surface fitting described above, and derive the very interesting result that the uncertainty in edge position and magnitude is highly dependent on the orientation of the edge with respect to the image coordinate axes.

1.3.1 Error in edge position

The Monga-Deriche 3-D modified Canny edge detector introduces two 1-D linear filters

$$\begin{aligned} S(x) &= \tfrac{\alpha}{4}(\alpha \mid x \mid +1)e^{-\alpha|x|} && \text{"smoothing" component,} \\ L(x) &= \alpha^2(1 - \alpha \mid x \mid)e^{-\alpha|x|} && \text{"Laplacian" component.} \end{aligned}$$

The image is convolved with the impulse response

$$D(x,y,z) = L(x)S(y)S(z) + S(x)L(y)S(z) + S(x)S(y)L(z),$$

which is an approximation to the local image Laplacian, and edges are then detected as the zero-crossings of D.

The extensions of Canny's model in 2D and 3D are respectively a straight line and a plane dividing the space into two areas of constant grey level (see figures 1 and 2).

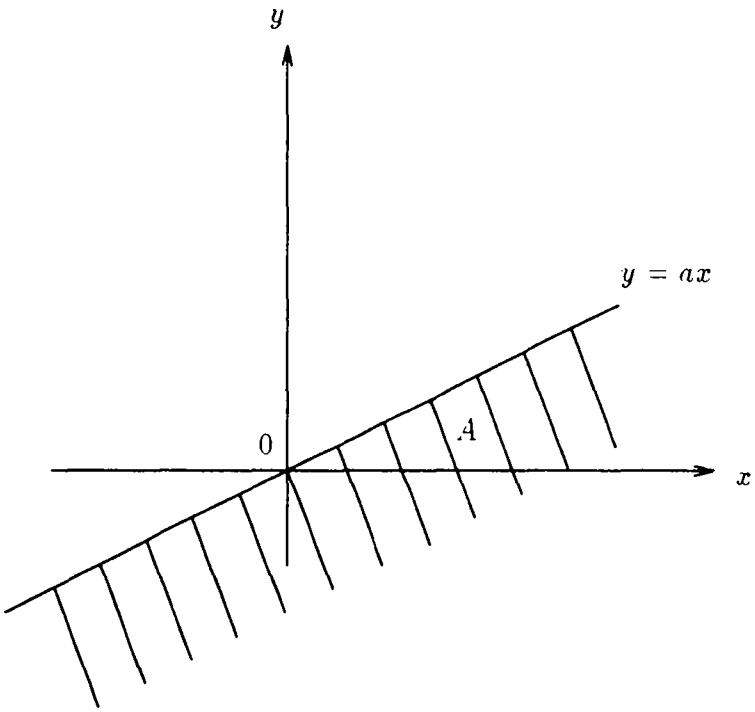

Figure 1: Ideal 2D edge

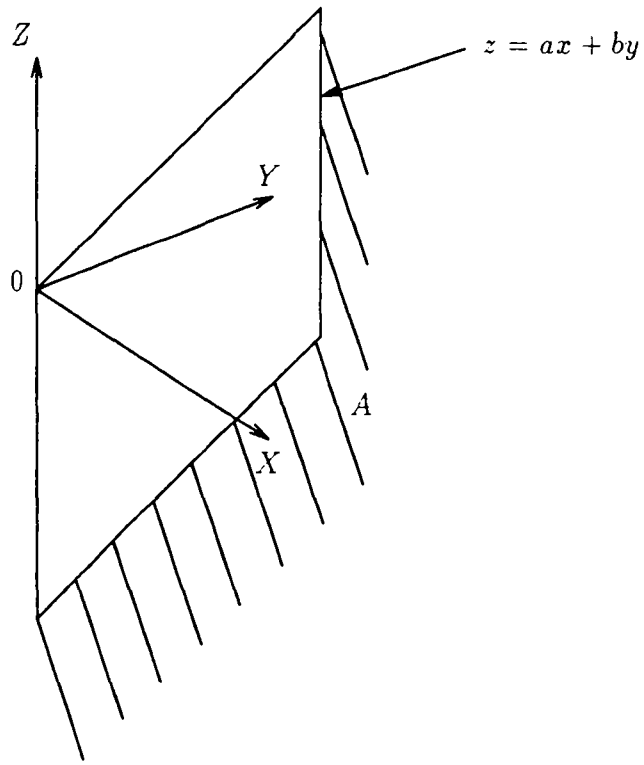

Figure 2: Ideal 3D edge

To compute the uncertainty in edge localization, we consider the "perfect 3-D edge" as being defined by a plane

$$H(x,y,z) = \begin{cases} 0 & \text{if } z > ax + by, \\ A & \text{if } z \leq ax + by, \end{cases}$$

with $a, b \geq 0$. A noisy edge

$$C(x,y,z) = H(x,y,z) + \eta(x,y,z)$$

is modeled by corrupting $H(x,y,z)$ with zero-mean additive Gaussian noise $\eta(x,y,z)$ (a generalization of the Canny 1D edge model).

We assume that the noise is separable into $\eta(x,y,z) = \eta_1(x) + \eta_2(y) + \eta_3(z)$ where the η_i's are independent zero-mean Gaussian noises, with covariance η_0^2. A point (x_0, y_0, z_0) is determined as being an edge point iff

$$\theta_0 = \theta(x_0,y_0,z_0) = \int\int\int C(x,y,z)D(x-x_0, y-y_0, z-z_0)\,dxdydz = 0,$$

i.e.,

$$\theta_0 = [C * D](x_0,y_0,z_0) = 0.$$

This convolution product can be rewritten as

$$\theta_0 = P_0 + N_0$$

where $P_0 = [H * D]_0$ (ideal edge component) and $N_0 = [\eta * D]_0$ (noise component). We first compute

$$P_0 = A\int\int\int_{z\leq ax+by} D(x_0 - x, y_0 - y, z_0 - z)\,dxdydz$$

and then approximate the result using a first order Taylor expansion of P_0 around the origin. We obtain[2]

$$P_0 = \simeq A\,(ax_0 + by_0 - z_0)\,\alpha^2\,f(a,b)$$

where

$$f(a,b) = \frac{(a^3b^3 + 3a^2b^3 + 3ab^3 + b^3 + 7a^2b^2 + 3a^3b^2 + 3ab^2 + 3a^3b + 3a^2b + a^3)(a^2 + b^2 - 1)}{(a+b)^3\,(a+1)^3\,(b+1)^3}.$$

[2]The computation was performed using the symbolic mathematics program Maple [CGGW85]. The listing of the result before evaluation of the symbolic expression is too long (30 pages) to be included in this publication.

Also, by using the fact that $\int_{-\infty}^{\infty} L(x)\,dx = 0$ and $\int_{-\infty}^{\infty} S(x)\,dx = 1$, we have

$$\begin{aligned} N_0 &= \int\int\int (\eta_1(x) + \eta_2(y) + \eta_3(z)) D(x_0 - x, y_0 - y, z_0 - z)\,dxdydz \\ &= \int \eta_1(x) L(x_0 - x)\,dx + \int \eta_2(y) L(y_0 - y)\,dy + \int \eta_3(z) L(z_0 - z)\,dz. \end{aligned}$$

Since $\theta_0 = 0$ at edge points, point (x_0, y_0, z_0) is an edge point when $P_0 + N_0 = 0$. Hence, for each detected edge point, we have $E[P_0^2] = E[N_0^2]$. Now, a simple computation shows that

$$\begin{aligned} E\left[N_0^2\right] &= \frac{3}{2}\alpha^3\eta_0^2, \\ E\left[P_0^2\right] &= E\left[(ax_0 + by_0 - z_0)^2\right] A^2\alpha^4 f^2(a,b), \end{aligned}$$

giving the standard deviation of the edge position as

$$\sigma\left[\frac{ax_0 + by_0 - z_0}{(a^2 + b^2 + 1)^{\frac{1}{2}}}\right] = \sqrt{\frac{3}{2\alpha}}\frac{\eta_0}{A} h(a,b), \tag{5}$$

where

$$h(a,b) = \frac{1}{f(a,b)\sqrt{a^2 + b^2 + 1}}.$$

Equation (5) provides a quantitative estimate of the quality of edge localization. As in the 1-D case, this shows that a large value of α yields good localization. The function $h(a,b)$ appears as a correction factor which depends on edge orientation, and which is minimum when the plane is parallel to one of the coordinates axes, i.e., when a and b both tend to zero or when one of them tends to infinity. In these three cases, $h(a,b) \to 1$. The maximum value of $h(a,b)$ is $h(1,1) = 128/21\sqrt{3} \simeq 3.52$, obtained with edge planes whose normal is parallel to vector $(1,1,-1)$. In this case, the localization is more that 3.5 times less accurate than in the previous cases! This clearly demonstrates the importance of accounting for the uncertainty in edge localization *as a function of edge spatial orientation.*

The 2-D case is covered by setting $b = 0$, which yields

$$h(a,0) = \frac{1}{f(a,b)\sqrt{a^2 + 1}} = \frac{(a+1)^3}{(a^2+1)^{\frac{3}{2}}}.$$

In this case, the minimum value of $h(a,0)$ is 1 for $a = 0$ or $a \to \infty$ (edges parallel to the coordinate axes). The maximum value (producing the worst localization) is $2\sqrt{2} \simeq 2.83$ and is again obtained for $a = 1$ (see figure 3).

1.3.2 Uncertainty of gradient magnitude

As we showed above, it is important to estimate the value of A to compute the accuracy of edge localization and to estimate the uncertainty in the edge direction. Information on both

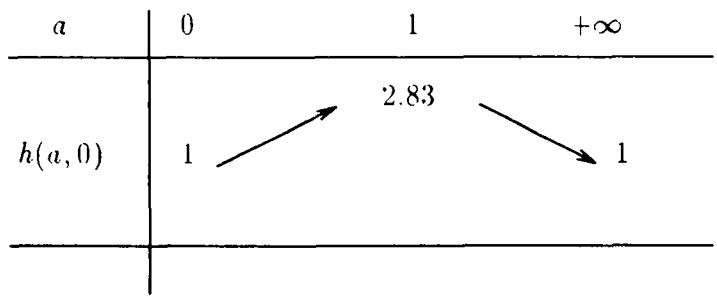

Figure 3: Change of monotony of $h(a,0)$

is provided by determining the edge gradient, which is computed by convolving the image with three derivative filters

$$\begin{aligned} g_1(x,y,z) &= d(x)S(y)S(z), \\ g_2(x,y,z) &= S(x)d(y)S(z), \\ g_3(x,y,z) &= S(x)S(y)d(z) \end{aligned}$$

where $S(x)$ is the previously defined smoothing filter, and $d(x)$ is the derivative filter

$$d(x) = c\,x\,e^{-\alpha|x|},$$

where c is a constant. For the ideal edge H,

$$\begin{aligned} G_{x_0} &= G_x(x_0,y_0,z_0) = [H * g_1]_0, = \int\int\int H(x,y,z)g_1(x_0-x, y_0-y, z_0-z)\,dxdydz, \\ G_{y_0} &= [H * g_2]_0, \\ G_{z_0} &= [H * g_3]_0, \end{aligned}$$

give the gradient vector $\mathbf{G_0} = (G_{x_0}, G_{y_0}, G_{z_0})^t$, whose direction is orthogonal to the contour, i.e., $\mathbf{G_0} \times (a,b,-1)^t = 0$. However, the gradient magnitude thus obtained again depends on the orientation of the contour, and is given by[3]

$$\|\mathbf{G_0}\| = A\,g(a,b),$$

where $g(a,b)$ is the ratio of two polynomials in a and b,

$g(a,b) =$

$$\frac{(ab+b+a)(b^4(a+1)^2+b^3(3a^3+8a^2+8a+3)+b^2(a^4+8a^3+15a^2+8a+1)+b(2a^4+8a^3+8a^2+2a)+a^4+3a^3+a^2)\sqrt{a^2+b^2+1}}{(b+1)^3(a+b)^3(a+1)^3}$$

[3]For a point lying on the contour, i.e., $z_0 = ax_0 + by_0$.

It is interesting to study how $g(a,b)$ varies with direction (a,b). It has a maximum value of 1 for a contour plane parallel to one of the coordinate system axes, i.e., when a and b both tend to zero or when one of them tends to infinity, and takes its minimum value 0.85 for $(a,b) = (1,1)$.

If we set $b = 0$, we have a 2-D contour, and in this case

$$g(a,0) = \frac{(a^2+3a+1)\sqrt{a^2+1}}{(a+1)^3}.$$

Again, the maximum value of 1 is obtained for $a = 0$ or $a \to \infty$. The minimum value $g(1,0) \simeq 0.88$ is obtained for $a = 1$(see figure 4).

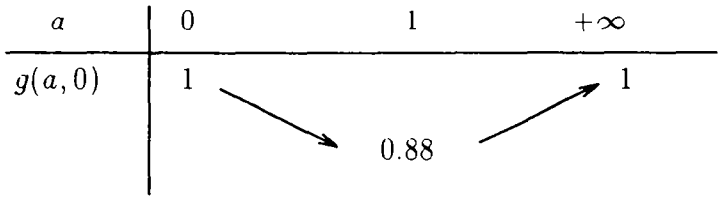

Figure 4: Change of monotony of $g(a,0)$

1.3.3 Practical computation of uncertainty

Position uncertainty

It is now possible to include the correct values of a, b and A in Eq. (5) of the standard deviation of edge localization. This is done by computing first the gradient vector $\mathbf{G} = (G_x, G_y, G_z)$, giving

$$(a,b) = (G_x/G_z, G_y/G_z).$$

We then compute the gradient magnitude $\|\mathbf{G_0}\|$, and divide it by the correcting factor $g(a,b)$ to obtain $A = \|\mathbf{G_0}\| \,/\, g(a,b)$. Putting these values back into Eq. (5), we obtain the standard deviation of edge localization

$$\sigma_\perp = \sqrt{\frac{3}{2\alpha}} \frac{\eta_0}{\|\mathbf{G_0}\|} q(a,b)$$

where

$$q(a,b) = g(a,b)\, h(a,b).$$

This covariance is finally included in a covariance matrix expressed in a coordinate system $(\mathbf{I},\mathbf{J},\mathbf{K})$ *attached to the edge*, i.e., such that the $\mathbf{K}$ axis is parallel to $(a,b,-1)^t$, and the $\mathbf{I}$

and **J** axes are parallel to the edge plane. This covariance matrix has the form

$$\Sigma_\perp = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & {\sigma^2}_\perp \end{pmatrix}.$$

The important fact here is that the covariance ${\sigma^2}_\perp$ ranges between a minimum value σ_{min} and a maximum value σ_{max} as a function on the 3-D orientation of the edge, and that the ratio $\sigma_{min}/\sigma_{max}$ is close to 9.6!

Knowing the orientation of the **I**, **J** and **K** axes, we can eventually express $\Sigma_\perp$ in the $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ coordinate system of the image as

$$\mathbf{\Sigma}_1 = \mathbf{R}_1 \mathbf{\Sigma}_\perp \mathbf{R}_1^t,$$

where $\mathbf{R}_1$ is as in §1.2.3. Finally, digitization noise is taken into account by computing

$$\mathbf{\Sigma}_2 = \mathbf{\Sigma}_1 + \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix},$$

where σ_x, σ_y and σ_z are directly proportional to voxel size in the **I**, **J** and **K** directions respectively.

Normal uncertainty

For the computation of the uncertainty of the surface normal direction, i.e., of its coordinates, recall that each component of $\mathbf{G}_0$ is obtained by convolving H with $g_i(x, y, z)$ for $i = 1, 2$ or 3. The covariance of each component of $\mathbf{G}_0$ is

$$\mathbf{W}_{\mathbf{G}_x} = \mathbf{W}_{\mathbf{G}_y} = \mathbf{W}_{\mathbf{G}_z} = \frac{3}{2}\alpha\eta_0^2.$$

Now, we divide $\mathbf{G}_0$ by $g^2(a, b)$ to compute a correctly scaled gradient vector, and this vector is used as the surface normal estimate denoted $\mathbf{n}_i$ in §1.2. Therefore, the covariance attached to this normal vector is

$$\mathbf{W}_{\mathbf{n}_i} = \frac{3\alpha\eta_0^2}{2g^2(a, b)}\mathbf{I}_3.$$

This is the covariance that was used in the experiments. Note that the computation applies to a point located exactly on the ideal edge. In fact, since we have estimated the uncertainty on edge localization, it should be possible to bound the error in position, and also to compute an upper bound on $\mathbf{W}_{\mathbf{n}_i}$. This will be done in the near future.

1.3.4 What about other edge detection operators ?

The same computation scheme could be used for any linear edge detection operator. Notice that the behaviour of isotropic operators such as Gaussian filter and its derivative does not depend on the edge orientation. For Canny's operator (first derivative of a Gaussian) the uncertainty in the 1D,2D and 3D cases are the same up to a multiplicative constant due to the noise. If we compare for all edge orientations the Gaussian and the Deriche filters, we obtain equivalent performances (the Deriche filter is better for an edge orientation close to X,Y,Z axis and Gaussian is better for an edge orientation along diagonals). We choose the Deriche filter because its recursive implementation yields a substancial saving in computing time. This is particularly important for 3D medical images where the huge amount of data makes the algorithmic complexity and the storage requirements key points for edge detection.

2 From curvatures to typical curvature features

2.1 Introduction

For each edge point, the previous section determines the Gaussian and mean curvatures, principal curvature directions, and the corresponding covariance matrices. Note that the scale is defined by the size of the neighbourhood used to fit the local geometric model. In this section we deal with the extraction of more global curvature features from the local curvature information.

2.2 Local curvature maps

A practical way of characterizing the behaviour of the curvature in the neighbourhood of a point is to define local curvature maps (technically the pullback of the field onto the tangent plane as used in [SZ] for the computation of the direction field index). Given a point P and its tangent plane defined by $(\mathbf{P}, \mathbf{Q}, \mathbf{N})$. Let V be the intersection of a sphere whose center is P and radius r with the set of the edge points (this defines a neighbourhood of P), and let W be the orthogonal projection of the points of V onto the tangent plane. At each point of W we attach the curvatures of its corresponding points in V. The size of V could be determined using the distance of the points to the tangent plane and the angle between the gradient at a point and the gradient at point P. We thus define a map characterizing the behaviour of the curvatures around P.

2.3 Smoothing curvatures

For a surface continuous up to second order, Gaussian and mean curvatures are locally continuous, but due to noise and to imperfections of the algorithms, we do not always obtain a coherent/continuous curvature field. It is thus necessary to use a refinement procedure to improve its smoothness [SZ90], and a natural idea is to derive a continuity model for the curvatures from the local geometric model of the surface as given by Eq.(1. Since the mean curvature at a point $P : (p, q)$ is a polynomial whose second order Taylor expansion around $(0, 0)$ is only composed of quadratic terms, it is appropriate to choose a second order polynomial with no linear terms as our continuity. We now have essentially the same situation as in §1.2, with the difference that the underlying function represents mean curvatures rather than distances from the local tangent plane. We proceed similarly, first by fitting the mean curvature model in the neighbourhood of P, and then replacing the curvature value at P by the value corresponding to the best fit.

The best fit could be defined by a classical least mean squares method, however the problem of least mean squares methods is that a few outlyers could considerably contaminate the results. To avoid this drawback we propose utilizing a Least Median Squares estimator (M-estimator) [PM90, JJ90]. The principle of this kind of estimator is to find a solution fitting only with a given percentage of the data. Notice that M-estimators could also be applied for the local geometric fitting described in §1.2. Recent results have shown that covariance matrices can also be taken into account by such methods [TSJ90]. The main problem of this kind of algorithms is the computation time which may be prohibitive.

2.4 Extracting extrema curvature lines

From the local curvature images we can extract, for instance, the maxima of the mean curvature in the direction of the curvature gradient. This may be done similarly to the classical extraction of the maxima of the gradient magnitude in the gradient direction in 2D edge detection methods [Der87, Can86].

Let $C(P)$ be the local mean curvature map attached to point P, and let $G(P)$ be the gradient of the mean curvature obtained, for instance, by applying the 2D gradient Deriche operator to the curvature map. We compare the value of the mean curvature along the straight line defined by P and $G(P)$ and retain P if its mean curvature is a local maximum along this direction. Thus we obtain local curvature extrema candidates. To remove false extrema, we perform a 3D hysteresis thresholding using the mean curvature already determined. This is done analogously to the thresholding of the local gradient extrema in the edge detection scheme described in [MDR91].

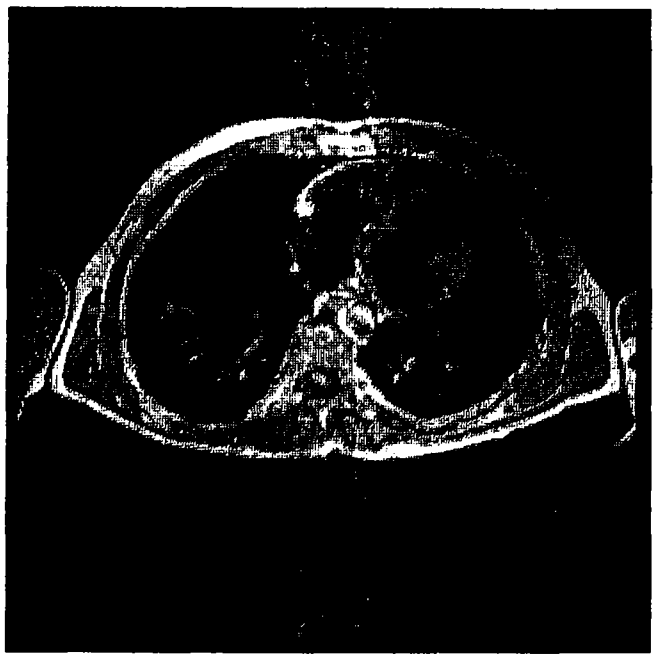

Figure 5: Original MR image

3 Results

3.1 Extraction of 3-D Edges

We performed three dimensional edge extraction, with the methods presented above, and then hysteresis thresholding, using first the original gradient magnitude $\mathbf{G}_0$, and then the correctly scaled gradient magnitude $\mathbf{G}_0/g(a,b)$. Figure 5 shows a cross section of a 3-D medical Magnetic Resonace Image of a body displayed at the level of the heart. Figure 6 shows the result of hysteresis thresholding on the original gradient magnitude, while figure 7 shows the results obtained with the correctly scaled gradient magnitude. The low and high thresholds used are exactly the same for both images. Figure 10 shows the difference between the two results, illustrating the advantage of using the corrected value of gradient magnitude. We see that an important piece of contour is ignored if we do not correct the gradient magnitude.

3.2 Curvatures for synthetic data

We built a synthetic volume whose implicit equation is given by

$$z \leq ax^2 + by^2 + cx + dy + e$$

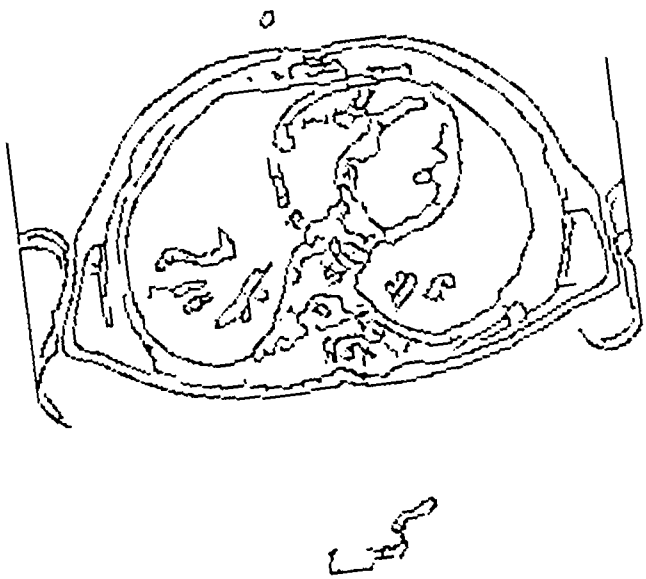

Figure 6: Hysteresis thresholding of edges with original gradient magnitude

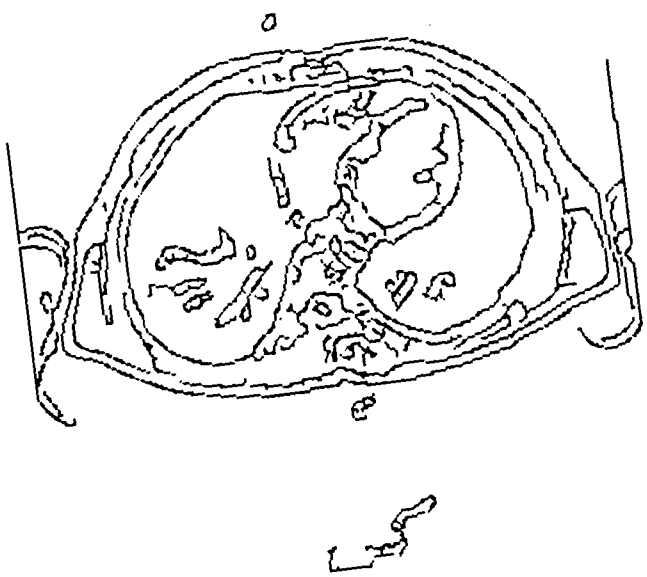

Figure 7: Hysteresis thresholding of edges with corrected gradient magnitude

Figure 8: Difference between previous edge images

with $a = 1/20$, $b = 0.25$, $c = -4$, $d = -12$, $e = 224$ (see figure 9 for the surface). This was done by creating a 3-D digital image with points of maximum (resp. minimum) intensity within (resp. outside of) the volume. We extracted 3-D edges with the algorithm described in §1.3. These edge points correspond to the surface of an ideal elliptic paraboloid.

First, note that there exists a single surface point T such that the equation of the *entire* surface takes the reduced form

$$z' = ex'^2 + 2fx'y' + gy'^2$$

when (x',y',z') are expressed in the local tangent plane coordinate system attached to point T. This point is the vertex of the paraboloid and its coordinates are given by

$$T = (\frac{-c}{2a}, \frac{-d}{2b}, \frac{-c^2}{4a} - \frac{d^2}{4b} + e).$$

For this particular point, the local quadric approximation is a *global* one, and the ideal parameters are $e = 2a$, $f = 0, g = 2b$. It is therefore possible to estimate these parameters with all the detected surface points (about 350 points). In this case the convergence is excellent towards the exact values. We show in table 1 the results obtained with a smaller but still rather large number of edge points (86). We show successively the results obtained for a least squares estimation with points only (measurement equation E1), then adding normals (measurement equations E2–E3). Finally, we show the results obtained when uncertainty on positions and normals is taken into account, following the computations of the previous

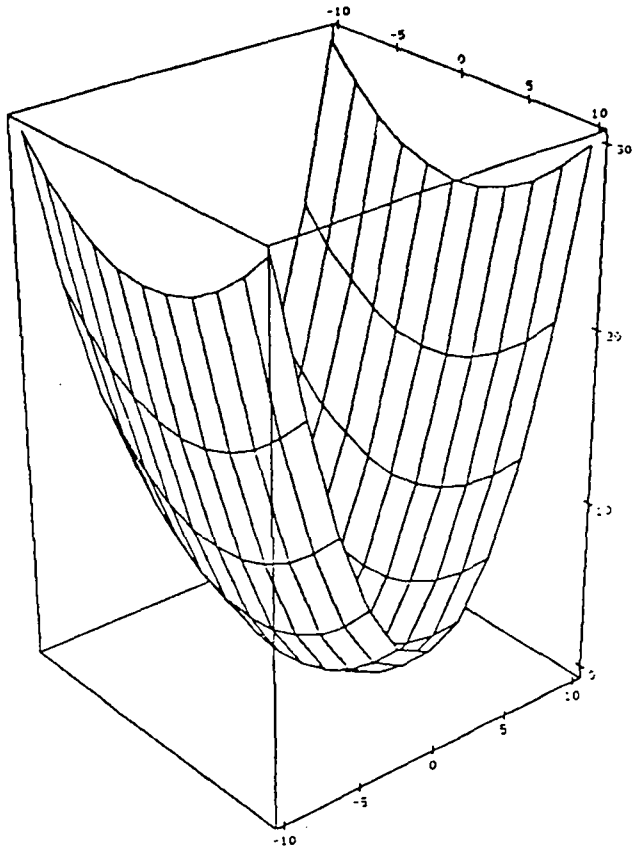

Figure 9: Elliptic parabolic showing surface shape of the synthetic volume.

sections. It is easy to check that not only is the estimate obtained much more accurate, but also that the computed standard deviation σ on the error estimation is perfectly coherent with the observed error.

At other points P on the surface, we applied our local quadric approximation with smaller neighborhoods (containing about 50 points). The size of the neighborhood is both controlled by limiting the angle between the neighbors' normal and P's normal, and also the distance between neighbors and P. We used the local approximation to compute locally the Gaussian and mean curvatures (C_g and C_m). In table 2, we show the results obtained for this computation, with the previous three methods. Here again, the results are much more

Point coordinates	Ideal	Positions	+ Normals	+ Uncertainty	Predicted σ
	$e = 0.1$	− 0.0037	0.0172	0.102	0.011
(40,24,1)	$f = 0$	0.0000	0.000	− 0.003	0.008
86 Neighbors	$g = 0.5$	0.324	0.360	0.494	0.02

Table 1: Estimation of the parameters (e, f, g) of the local quadrics

Point coordinates	Ideal	Positions	+ Normals	+ Uncertainty	Predicted σ
(40,24,1)	$C_g = 0.05$	− 0.0012	0.0062	0.0505	0.014
86 Neighbors	$C_m = 0.3$	0.16	0.16	0.298	0.023
(47,24,2)	$C_g = 0.022$	0.089	0.063	0.0157	0.008
46 Neighbors	$C_m = 0.23$	0.333	0.297	0.2024	0.030
(45,26,3)	$C_g = 0.010$	− 0.001	− 0.0003	0.018	0.009
54 Neighbors	$C_m = 0.12$	0.053	0.107	0.143	0.018
(36,22,2)	$C_g = 0.010$	0.013	0.018	0.012	0.006
50 Neighbors	$C_m = 0.122$	0.133	0.172	0.127	0.015
(39,23,1)	$C_g = 0.031$	0.011	0.023	0.017	0.023
28 Neighbors	$C_m = 0.22$	0.10	0.25	0.21	0.062
(38,28,5)	$C_g = 0.004$	0.009	0.010	0.006	0.003
50 Neighbors	$G_m = 0.07$	0.105	0.11	0.09	0.01

Table 2: Estimation of gaussian and mean curvatures C_g and C_m

accurate with the last approach, and the output covariance agrees almost perfectly well with the observed errors.

3.3 Curvatures for real data

The goal of the experiments reported here is to show the stability of the mean and Gaussian curvature provided by our algorithm. We consider two 3D scanner images of the same vertebra but taken at two different positions A and B (see figures 9 and 10). We perform the sequence of processes (edge detection, approximation of the curvatures) for a part of the vertebra. We obtain Gaussian and mean curvature images corresponding to positions A and B. (see figures 11 to 13). We compute the rigid transformation (rotation and translation) from B to A : B-A using an algorithm described in [BK89]. We compare mean and Gaussian curvature images for the position A to the mean and Gaussian curvature images for the position B transformed by B-A (see figures 11 to 13). This allows a valid comparison of the curvatures computed for positions A and B. The stability of the results obtained for the two positions shows clearly the robustness of our algorithms.

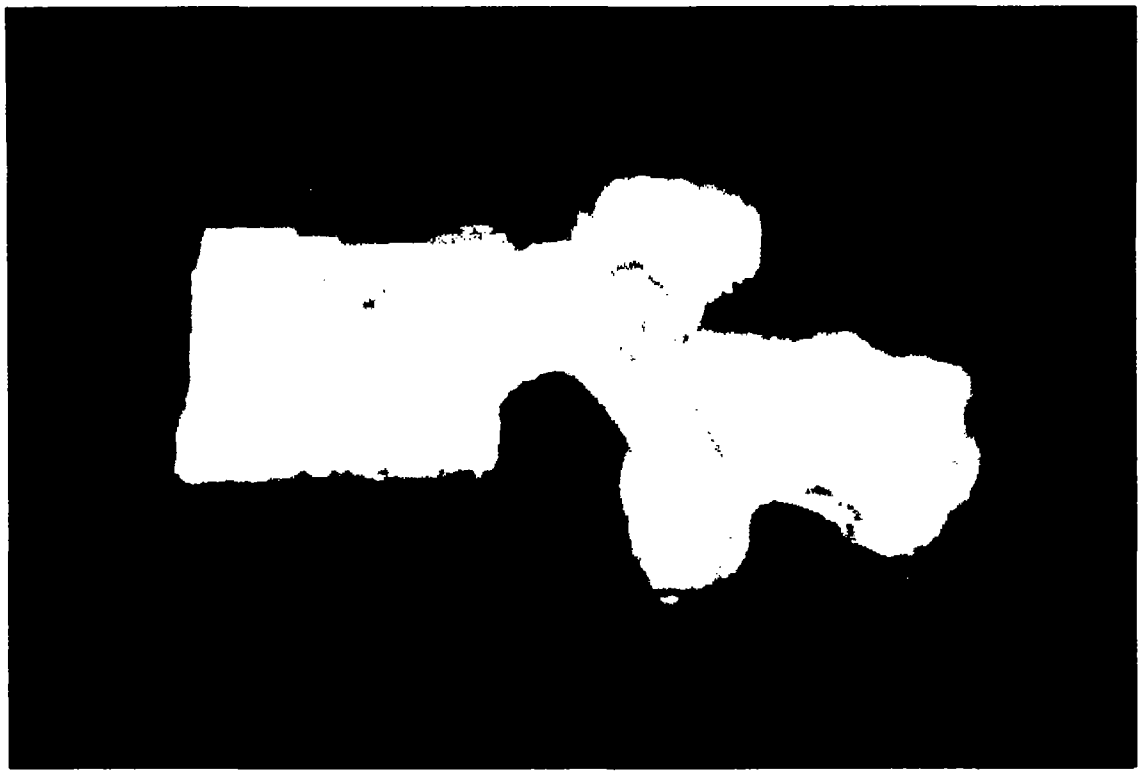

Figure 10: Perspective view of the 3D image of the vertebra for position A

3.4 Typical curvature features for synthetic and real data

At the time of the conference we will present some results for the smoothing of the curvatures using Median Least Squares and also for the determination of the extrema of the mean curvature in the principal direction of curvature. At that date, these algorithms have been implemented and tested on simple synthetic data.

4 Conclusion

Our main objective was to develop robust and reliable tools useful for modeling and analyzing surfaces of 3-D objects. In this paper we showed the importance of a careful quantitative analysis of the various sources of uncertainty for computing second order derivative features (mean and Gaussian curvatures) on a discrete surface.

We analyzed *quantitatively* the uncertainty in edge position, orientation and magnitude produced by the multidimensional (2-D and 3-D) versions of the Monga-Deriche-Canny recursive separable edge-detector. We showed for instance that, depending on the orientation, the position uncertainty may vary with a ratio larger than 2.8 in the 2-D case, and 3.5 in the 3-D case. We showed that to estimate edge magnitude, the gradient magnitude must be

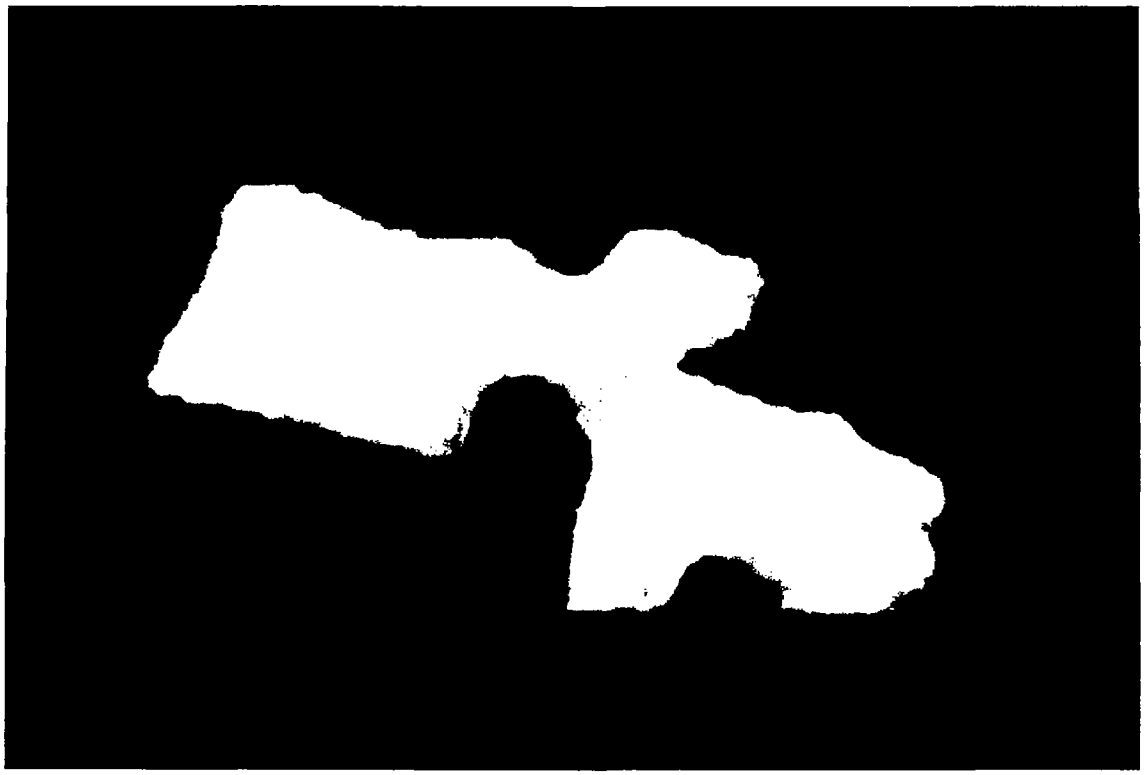

Figure 11: Perspective view of the 3D image of the vertebra for position B

corrected by a factor which depends on the orientation, and whose relative variation is close to 20%.

Then, we revisited the algorithm initially proposed by Sander and Zucker for locally estimating the curvature of a discrete 3-D surface, and we modified the original measurement equations and proposed an optimal estimation scheme to account for the previously computed uncertainties and corrections.

We tested the corrected edge detector on 2-D and 3-D medical images and showed the importance of the corrected edge magnitude for edge detection. We also tested the surface modeling algorithm on discrete 3-D objects — not only are the results obtained more accurate, but the computed measure of uncertainty attached to the results agrees extremely well with the true one.

We also show how to use these curvatures to determine typical curvature features on which registration and/or tracking procedures can robustly rely.

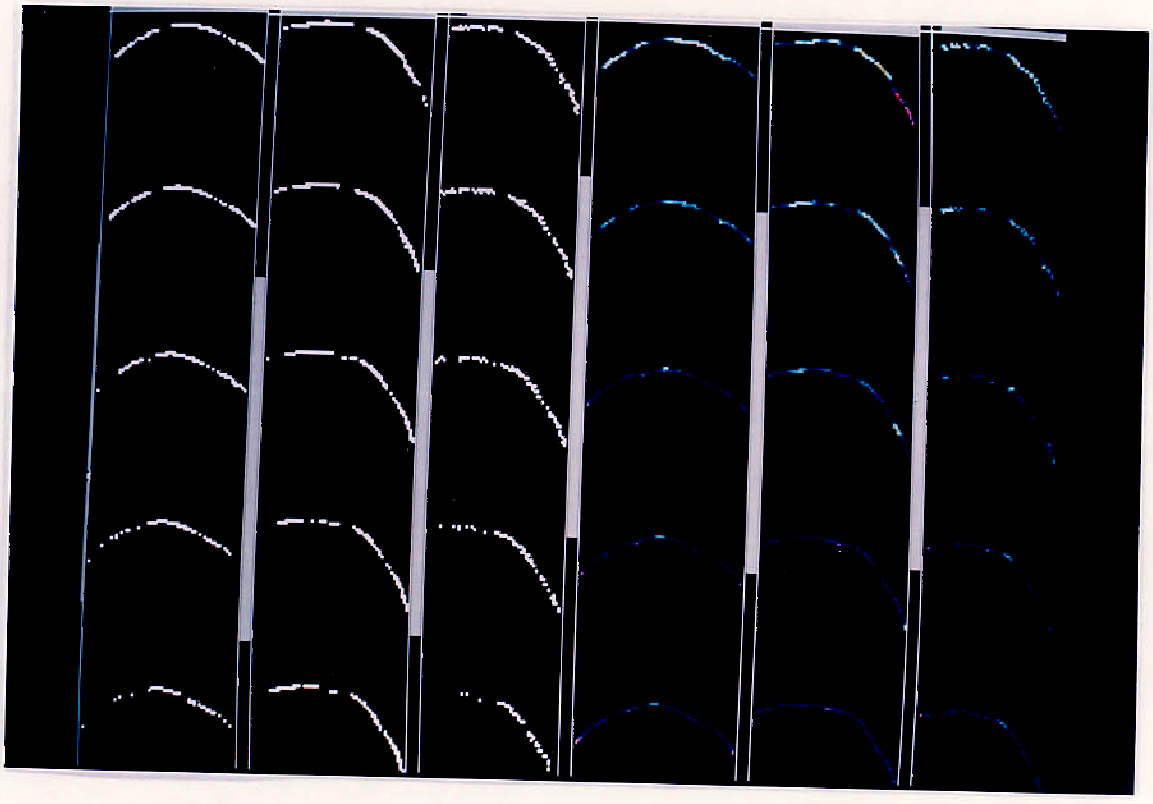

Figure 12: First columm : 3D image (5 successsive planes) corresponding to mean curvature for position A ; second columm : 3D image corresponding to mean curvature for position B transformed by B-A ; third columm : 3D image corresponding to mean curvature for position B ; Fourth, fifth and sixth columm : same as the three precedents but for Gaussian curvatures. *For these results the size of the neighbourhood used for the local approximation is 5*

Figure 13: Same as the columms 1 and 2 of the precedent figure but using a neighbourhood of size 10 for the local approximation

Figure 14: Same as the precedent figure but for Gaussian curvature

Acknowledgements

We thank Pr. Bittoun of the Kremlin Bicetre Hospital in Paris for supplying the medical images. Nathalie Gaudechoux provided substantial help in the preparation of this manuscript.

References

[AB+90] N. Ayache, J.D. Boissonnat, , L. Cohen, , B. Geiger, J. Levy-Vehel, O. Monga, and P. Sander. Steps toward the automatic interpretation of 3d images. In *Proceedings of the NATO Advanced Research Workshop on 3D Imaging in Medicine*, Travemünde, June 1990. NATO ASI Series, Springer-Verlag.

[Aya91] N. Ayache. *Artificial Vision for Mobile Robots – Stereo-Vision and Multisensory Perception*. MIT Press, Boston, 1991.

[BJ88] Paul J. Besl and Ramesh C. Jain. Segmentation through Variable-Order surface fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

PAMI-10(2):167–192, March 1988.

[BK89] Ruzena Bajcsy and Stane Kovacic. Multiresolution elastic matching. *Computer Vision Graphics and Image Processing*, 46:1–21, 1989.

[Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[CGGW85] Bruce W. Char, Keith O. Geddes, Gaston H. Gonnet, and Stephen W. Watt. *Maple Reference Manual, 4th edition.* Dept. of Computer Science, University of Waterloo, Waterloo, Ont., 1985.

[dC76] Manfredo P. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Englewood Cliffs, 1976.

[Der87] Rachid Deriche. Using Canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, pages 167–187, 1987.

[HZ83] Robert A. Hummel and Steven W. Zucker. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:267–287, 1983.

[JJ90] S. Bataouche J.M. Jolion, P. Meer. Range image segmentation by robust clustering. Technical Report CAR-TR-500 CS-TR-2456, University of Maryland, College Park, Maryland 20742, April 1990.

[Lue69] David G. Luenberger. *Optimization by Vector Space Methods.* Wiley, New York, 1969.

[MD89] Olivier Monga and Rachid Deriche. 3d edge detection using recursive filtering. In *Conference on Vision and Patern Recognition*, San Diego, June 1989. IEEE.

[MDMC90] Olivier Monga, Rachid Deriche, Gregoire Malandain, and Jean-Pierre Cocquerez. Recursive filtering and edge closing: two primary tools for 3d edge detection. In *First European Conference on Computer Vision (ECCV), also to appear in Image and Vision Computing*, Antibes, April 1990.

[MDR91] Olivier Monga, Rachid Deriche, and Jean-Marie Rocchisani. 3d edge detection using recursive filtering: Application to scanner images. *to appear in Computer Vision Graphic and Image Processing*, January 1991.

[PM90] A. Rosenfeld P. Meer, D. Mintz. Least median of squares based robust analysis of image structure. Technical Report CAR-TR-490 CS-TR-2428, University of Maryland, College Park, Maryland 20742, March 1990.

[SZ] Peter T. Sander and Steven W. Zucker. Singularities of principal direction fields from 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. To appear. Available as Technical Report CIM-88-7, McGill Research Center for Intelligent Machines, McGill University, Montréal.

[SZ87] Peter T. Sander and Steven W. Zucker. Tracing surfaces for surfacing traces. In *Proceedings of the First International Conference on Computer Vision*, pages 241–249, London, June 1987.

[SZ90] Peter T. Sander and Steven W. Zucker. Inferring surface trace and differential structure from 3-D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9), September 1990.

[TSJ90] Arun Tirumalai, Brian Schunck, and Ramesh C. Jain. Robust dynamic stereo for incremental disparity map refinement. In *Proceedings of the Workshop on Robust Computer Vision*, 1990.

[ZH81] S.W. Zucker and R.M. Hummel. A three-dimensional edge operator. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(3):324–331, May 1981.

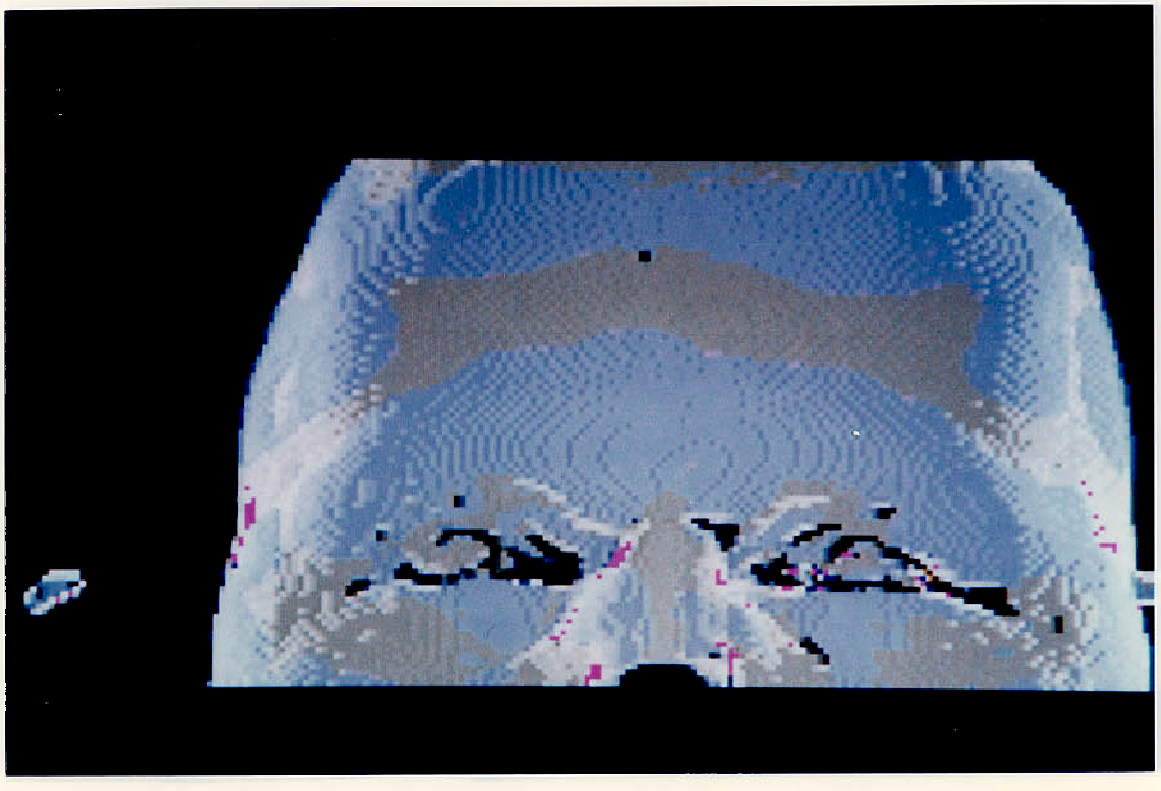

Figure 15: Sign of the Gaussian curvature for a NMR image of the head where we have extracted the part corresponding to the face ; hyperbolic points are green, points where the Gaussian curvature is zero are red, elliptic points are grey

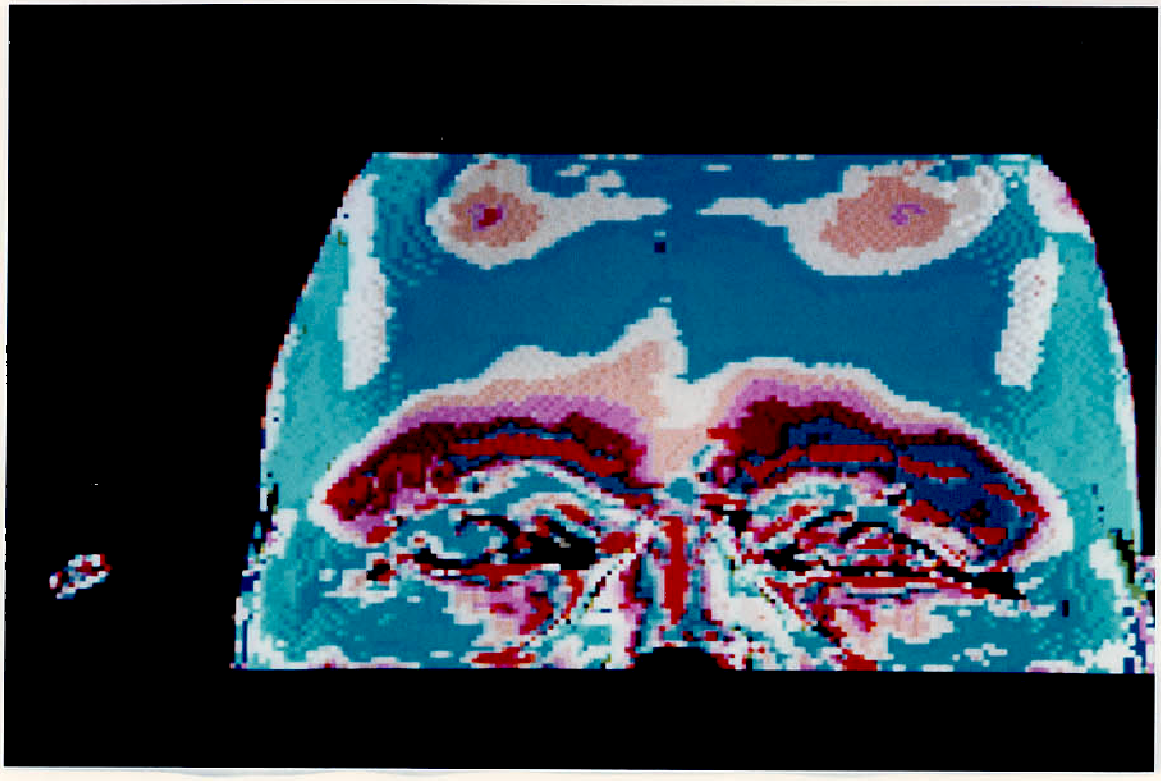

Figure 16: Maximum curvature map ; the extrema of the maximum curvature in the maximum curvature direction are colored in red (see next figure)

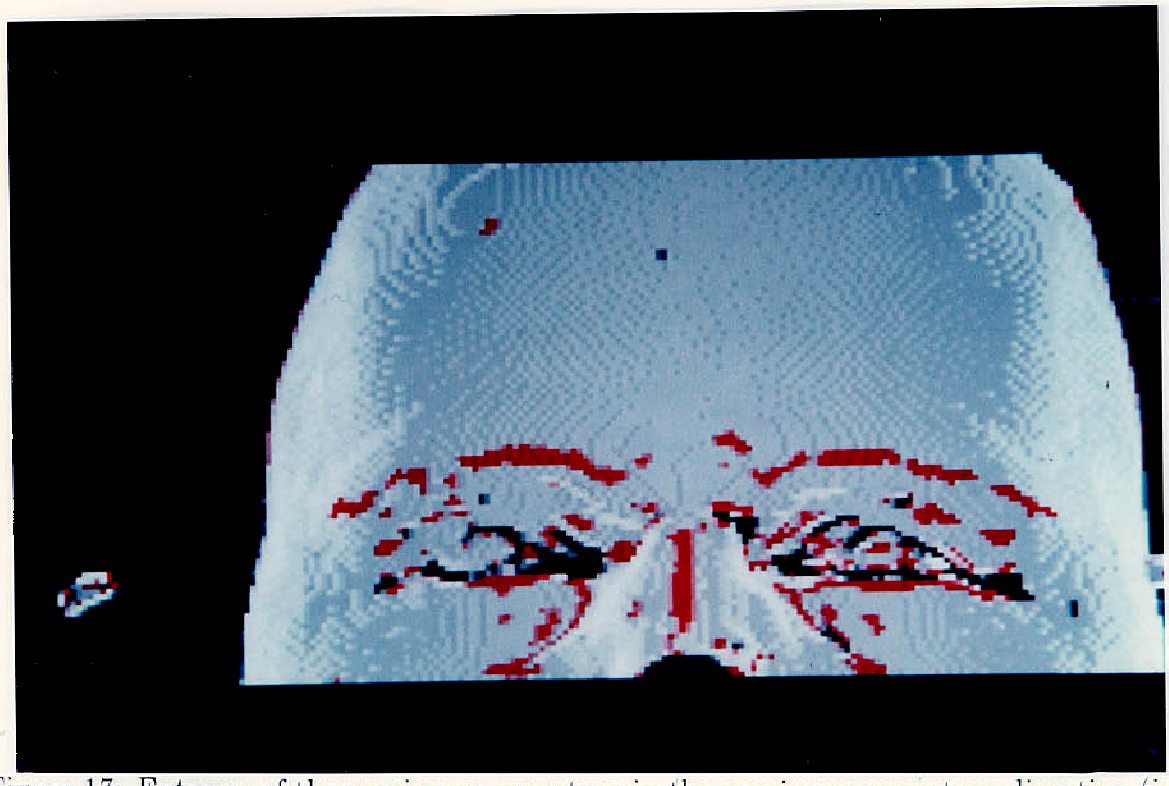

Figure 17: Extrema of the maximum curvature in the maximum curvature direction (in red)

Acknowledgement

This work has been partially supported by Ge-Cgr and some of the presented images were acquired in collaboration with the Advanced Image Processing Group of GE-CGR in Buc, France. We shall give the details of this acquisition process, and a thorough description of the results very soon in a forthcoming joint paper Inria-Ge-Cgr.

Imprimé en France
par
.l'Institut National de Recherche en Informatique et en Automatique.

ISSN 0249-6399