

Using partial derivatives of 3D images to extract typical surface features

Olivier Monga, Serge Benayoun

▶ **To cite this version:**

Olivier Monga, Serge Benayoun. Using partial derivatives of 3D images to extract typical surface features. [Research Report] RR-1599, INRIA. 1992. inria-00074961

HAL Id: inria-00074961
https://inria.hal.science/inria-00074961

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

1 9 9 2

25ème anniversaire

N° 1599

Programme 4
Robotique, Image et Vision

USING PARTIAL DERIVATIVES OF 3D IMAGES TO EXTRACT TYPICAL SURFACE FEATURES

Olivier MONGA
Serge BENAYOUN

Février 1992

RR_1599

Using partial derivatives of 3D images to extract typical surface features

Olivier Monga
Serge Benayoun
INRIA, Domaine de Voluceau - Rocquencourt
BP. 105, 78153 Le Chesnay Cedex, France

INRIA research report January 1991, Programme 4,
submitted to CVGIP :Image Understanding

Abstract

3D edge detection in voxel images provides points corresponding to 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 tracking procedures can rely. Typically one must calculate second order differential characteristics of the surfaces such as maximum, mean and Gaussian curvatures.

The classical approach is to use local surface fitting, and therefore it faces the problem of establishing links between 3D edge detection and local surface approximation. To get rid of this question, we propose to compute the curvatures on the edge points from the partial derivatives of the image.

By assuming that the surface is defined by the iso-contours (ie the 3D gradient at an edge point corresponds to the normal to the surface) one can calculate directly the curvatures and characterize the local curvature extrema (ridge points) from the first, second and third derivatives of the grey level function. These partial derivatives can be computed using the operators of the edge detection.

In the more general case where the contours are not iso-contours (ie the gradient at an edge point dœs not approximate the normal to the surface) the only differential invariants of the image are in R^4. This leads us to treat the 3D image as a hypersurface (a 3D dimensional manifold) in R^4. We set up the relationships existing between the curvatures of the hypersurface and the curvatures of the surface traced by the edge points. We express the maximum curvature at a point of the hypersurface with the second partial derivatives of the 3D image. We notice that it may be more efficient to smooth the data in R^4. Moreover this approach could also be used to detect corners or vertices.

We present experimental results obtained on real data (X-scanner images) using these two methods. As an application we extract the ridge line and show their stability using two 3D X-scanner images of a skull taken at different positions.

I Introduction

Three-dimensional edge detection in voxel images is used to locate points corresponding to surfaces of 3D structures [ZH81] [MDR91, MDM91]. The next stage is to characterize the local geometry of these surfaces in order to extract points or lines which may be used by registration, tracking and matching procedures [SZ90, BPYA85, PB85, Koe90]. Typically one must calculate second order differential characteristics of the surfaces such as the maximum, gaussian and mean curvature. The classical scheme is to use local surface fitting on the adjacency graph defined by the edge points [SZ87]. One faces the problem of linking surface elements detected by 3D edge operators and local surface approximation, which may be done taking uncertainty into account [MAS91].

In this paper we show how to compute the curvature of the surface defined by the edge points from the second partial derivatives of the image. The partial derivatives of the 3D image can be computed using the same operators as the ones used for edge detection [MDM91]. We will propose two methods.

By assuming that the surface is defined by an iso-contour (i.e. the 3D gradient at an edge point corresponds to the normal to the surface) one can compute directly the curvatures and characterize the local curvature extrema (ridge points will be make more precise later, in section IV) from the first, second and third derivatives of the grey level function.

In the more general case where the contours are not iso-contours (i.e. the gradient at an edge point does not approximate the normal to the surface) the only differential invariants of the image are in $\mathbf{R}^4$. This leads us to treat the 3D image as a hypersurface (a three dimensional manifold) in $\mathbf{R}^4$. We set up the relationships existing between the curvatures of the hypersurface and the curvatures of the surface traced by the edge points. We express the maximum curvature at a point of the hypersurface with the second partial derivatives of the 3D image. We notice that it may be more efficient to smooth the data in $\mathbf{R}^4$. Moreover this approach could also be used to detect corners or vertices. We notice that intuitively the curvatures of the hypersurface are related to both the local geometric properties of the surface and the grey level function. The same problem arises one dimension lower when dealing with the extraction of corners and vertices from 2D images using the curvature of the surface traced by the image [Nob88, DG90]. Using as a geometric model a 3D roof with a step edge and Deriche filter [Der87] to calculate the second partial derivative we link the curvatures of the hypersurface with the curvatures of the surface.

From these computations we obtain two efficient algorithms to compute the curvatures from the second partial derivative obtained by filtering with the same operators as used for edge detection. Compared to classical approaches using local surface fitting we obtain both better accuracy, and sub-

Des dérivées partielles des images 3D aux caractéristiques intrinsèques des surfaces

resumé

La détection de contours 3D dans des images de voxels fournit des points correspondant aux surfaces formant les structures 3D. L'étape suivante consiste a caractériser la géométrie locale de ces surfaces de manière à extraire des amères (points, lignes..) sur lesquels les algorithmes de fusion multicapteurs, de suivi et de mise en correspondance peuvent s'appuyer. Typiquement, on est amené à calculer des caractéristiques différentielles du second ordre des surfaces telles que les courbures maximum, Gaussiennes et moyenne.

L'approche classique est d'approximer localement par une surface, on est ainsi confronté au problème d'établir des liens entre la détection de contours 3D et l'approximation locale de surface. De manière à tourner cette difficulté, nous proposons de calculer les courbures aux points de contour a partir des dérivées partielles de l'image.

Si on suppose que la surface est définie par les iso-contours (ie le gradient 3D en un point de contour correspond a la normale à la surface) on peut calculer directement les courbures et caractériser les extrema locaux des courbures (points de crête) a partir des dérivées partielles première, seconde et troisième de la fonction des niveaux de gris. Ces dérivées partielles peuvent être calculées en utilisant le même type d'opérateurs que pour la détection de contours.

Dans le cas le plus général où les contours ne sont pas des iso-contours (ie le gradient en un point de contour n'approxime pas la normale à la surface en ce point) les seuls invariants différentielles de l'image sont dans R^4. Nous établissons les relations existant entre les courbures de l'hypersurface et les courbures de la surface définie par les points de contour. Nous exprimons la courbure maximum en un point de l'hypersurface avec les dérivées partielles d'ordres 1 et 2 de l'image 3D. Nous remarquons qu'il peut être plus intéressant de lisser les données dans R^4 De plus cette approche peut aussi être utilisée pour la détection de coins et de contours multiples.

Nous présentons des résultats expérimentaux obtenus sur des données réelles (images scanner-X) en utilisant ces deux méthodes. Comme application nous détectons les lignes de crête et nous montrons leur stabilité sur deux images 3D scanner-X d'un crâne prises à deux positions différentes. es

stancial saving in computational effort.

The paper is organized as follow. In the second section we perform the calculation of first, second and third partial derivatives of a 3D image. The third section deals with the direct computation of the curvatures using the first and second partial derivatives. In the fourth section we give a definition for ridge points and compute it using the first, second and third partial derivatives. The fifth section presents a practical algorithm based on the previous developement. In the sixth section we recall some basic notions from differential geometry about hypersurfaces in $\mathbb{R}^4$ [Spi71, KB76, Kob], and apply them to 3D images. In the seventh section we present the calculus linking the hypersurface curvatures and the surface curvatures. The eighth section presents a practical algorithm to extract ridge lines in a 3D image using hypersurfaces. The ninth section gives experimental results of these two approaches applied to X-ray scanner data of a skull.

II Computing partial derivatives

We use the 3D Deriche filters [Der87, MDM91] to compute the partial derivatives of image data. We set:

$$f_0(x) = c_0(1 + \alpha \mid x \mid)e^{-\alpha|x|}, \text{(smoothing operator)};$$
$$f_1(x) = -c_1 x \alpha^2 e^{-\alpha|x|}, \text{(first derivative operator)};$$
$$f_2(x) = c_2(1 - c_3\alpha \mid x \mid)e^{-\alpha|x|}, \text{(second derivative operator); and}$$
$$f_3(x) = c_4\frac{\mid x \mid}{x}(c_5 + 1 - \alpha c_5 \mid x \mid)e^{-\alpha|x|}, \text{(third derivative operator).}$$

These functions are used as convolution filters to provide the smoothed zero, first, second, and third derivatives of a function of a single variable. The normalization coefficients $c_0, c_1, c_2, c_3, c_4, c_5$ are chosen so that

$$\int_{-\infty}^{+\infty} f_0(x)dx = 1;$$

$$\int_{-\infty}^{+\infty} x f_1(x)dx = 1;$$

$$\int_{-\infty}^{+\infty} f_2(x)dx = 0 \text{ and } \int_{-\infty}^{+\infty} \frac{x^2}{2} f_2(x)dx = 1;$$

$$\int_{-\infty}^{+\infty} \frac{x^3}{6} f_3(x)dx = 1 \text{ and } \int_{-\infty}^{+\infty} x f_3(x)dx = 0.$$

These conditions ensure that convolution by the filters f_0, f_1, f_2, f_3 yield the correct derivatives when applied to polynomials. When the filters are to be applied in a discrete setting, similar formulas are used to find the

normalization coefficients. For example, for integer sampling, we have the formula for c_0:

$$\sum_{-\infty}^{+\infty} f_0(n) = 1.$$

Using discrete versions of the above formulas, we obtain the following values for the normalization coefficients for the case of discrete convolutions:

$$c_0 = \frac{(1-e^{-\alpha})^2}{1+2e^{-\alpha}\alpha - e^{-2\alpha}};$$

$$c_1 = \frac{-(1-e^{-\alpha})^3}{2\alpha^2 e^{-\alpha}(1+e^{-\alpha})};$$

$$c_2 = \frac{-2(1-e^{-\alpha})^4}{1+2e^{-\alpha}-2e^{-3\alpha}-e^{-4\alpha}};$$

$$c_3 = \frac{(1-e^{-2\alpha})}{2\alpha e^{-\alpha}};$$

$$c_4 = -\frac{(1-e^{-\alpha})^4((\alpha+1)e^{-\alpha}+\alpha-1)}{2\alpha^2 e^{-2\alpha}(1+e^{-\alpha})}; \text{ and}$$

$$c_5 = \frac{1-e^{-\alpha}}{(\alpha+1)e^{-\alpha}+\alpha-1}.$$

The value of α controls the degree of smoothing, and must still be specified in order to complete the specification of f_0, f_1, f_2, f_3.

Each function $f_{i+1}(x)$ is (by construction) proportional to the derivative of the previous function $f_i(x)$, except at $x = 0$, where the functions are not differentiable. Interestingly, however, the constants of proportionality are not necessarily unity, and in the case of our discrete filters, differ from one. See Deriche [Der89] for a more complete discussion.

Given an image function $I = I(x, y, z)$, separable filters are used to formally compute smoothed partial derivatives. Using subscripts to denote partial derivatives, we use the following approximations in order to compute smoothed partial derivatives:

$$I_x = (f_1(x)f_0(y)f_0(z)) * I,$$

$$I_{xx} = (f_2(x)f_0(y)f_0(z)) * I,$$

$$I_{xy} = (f_1(x)f_1(y)f_0(z)) * I,$$

$$I_{xyz} = (f_1(x)f_1(y)f_1(z)) * I,$$

$$I_{xxz} = (f_2(x)f_0(y)f_1(z)) * I,$$

$$I_{xxx} = (f_3(x)f_0(y)f_0(z)) * I.$$

These formulas give exact results when I is a polynomial in (x, y, z) but otherwise give approximate results. The filters can be implemented using recursive filtering [Der89, MDM91] or convolution masks. The most efficient implementation depends on the width of the filters and on the number of points where the curvature values are desired, i.e., surface edge points. (In the recursive implementation we perform the filtering for all the points of the image and we use only the result for the edge points.) In practice we use different values of α depending on the order of the derivative. In our experiments, we chose to set α to 1 for first order, to 0.7 for second order, and to 0.5 for third order derivatives. These choices are motivated by the fact that higher order derivatives require greater smoothing in order to obtain reasonable immunity from noise [VF92].

III Curvatures from partial derivatives

In this section, we give the formulas for computing curvatures using only the information from partial derivatives of the intensity data.

III.1 2D case

We first present the case in 2D in order to motivate the computations in 3D.

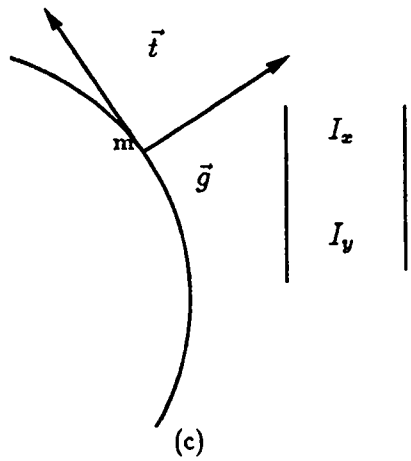


Figure 1:

Let (C) be a contour in an image $I(x, y, z)$, and we suppose that I is differentiable and that C is an iso-contour. We denote by $\vec{g}$ the gradient of I which will lie normal to the curve along C (see figure 1). We wish to compute the curvature k of (C) at point m.

Let $\vec{t}$ be the curve unit tangent vector to (C) at point m

$$\vec{g} \cdot \vec{t} = 0$$

Let s be the arclength parameter, we have :

$$\frac{d(\vec{g} \cdot \vec{t})}{ds} = \frac{d\vec{g}}{ds} \cdot \vec{t} + \vec{g} \cdot \frac{d\vec{t}}{ds} = 0 \quad (1)$$

The curvature k is obtained by :

$$\frac{d\vec{t}}{ds} = k\,\vec{n} = k\frac{\vec{g}}{\|\vec{g}\|} \quad (2)$$

where $\vec{n}$ is the curve unit normal vector (equal to $\frac{\vec{g}}{\|\vec{g}\|}$).

We must suppose here that the normal vector field is regular over the whole image to apply the chainrule. That is an approximation because the contour is not defined everywhere. We also **identify the normal vector field to the gradient vector field**, which is correct in the case of iso-contours. This field is regular because issued from a filtering technique. We notice that this interpretation problem does not arise in the local contour fitting algorithm where only the contour points are taken into account, or in the hypersurface algorithm where the whole image is the trace of a surface in $\mathcal{R}^4$. Further assuming that $\vec{g}$ is defined everywhere and is a C^2 mapping from $\mathbf{R}$ to $\mathbf{R}^2$, the chainrule gives :

$$\frac{d\vec{g}}{ds} = \frac{\partial \vec{g}}{\partial x}\frac{dx}{ds} + \frac{\partial \vec{g}}{\partial y}\frac{dy}{ds} = H\vec{t} \quad (3)$$

where H is the Hessian of the grey level function $I(x, y)$:

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

By combining equations (1), (2) and (3) we obtain

$$k = \frac{-t^T H t}{\|\vec{g}\|}$$

The tangent $\vec{t}$ can be computed simply as :

$$\|\vec{g}\|\vec{t} = \begin{pmatrix} -I_y \\ I_x \end{pmatrix}$$

III.2 3D case

We consider a surface $\sum$ defined as an iso-surface of $I(x,y,z)$. Let $\vec{t}$ be any unit vector in the tangent plane T_m at a point m and $\vec{g}$ the normal to the surface at m (see figure 2). We wish to compute the curvature of $\sum$ in the direction $\vec{t}$ which we denote by $k_{\vec{t}}$.
Using again the fact that

$$\vec{g} \cdot \vec{t} = 0$$

in $\sum$, and taking a directional derivative in the direction $\vec{t}$, we obtain as in the 2D case :

$$L_{\vec{t}}(\vec{g} \cdot \vec{t}) = t^T H t + k_{\vec{t}} \vec{g} \cdot \vec{n}$$

where $L_{\vec{t}}$ is the Lie derivation operator and H is the Hessian of the grey level function $I(x,y,z)$:

$$H = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix}$$

Since

$$\vec{n} = \frac{\vec{g}}{\|\vec{g}\|}$$

Using :

(4)
$$\frac{d\vec{g}}{ds} = H\vec{t}$$

and

(5)
$$\frac{d\vec{t}}{ds} = k_{\vec{t}}\vec{n} = k_{\vec{t}}\frac{\vec{g}}{\|\vec{g}\|}$$

It comes :

$$k_{\vec{t}} = -\frac{t^T H t}{\|\vec{g}\|}$$

The principal curvatures and the principal curvature directions can be computed by searching the directions $\vec{t}$ for which $k_{\vec{t}}$ is an extremum. This may be done using the following calculus :

$$\vec{g} = \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix}$$

Let $\vec{a}$ and $\vec{b}$ form an orthonormal basis of the tangent plane :

$$\|\vec{a}\| = \|\vec{b}\| = 1$$

$$\vec{a} \cdot \vec{g} = \vec{b} \cdot \vec{g} = 0;$$

$$\vec{t} = \cos\theta\ \vec{a} + \sin\theta\ \vec{b}$$

If we derive $k_{\vec{t}}$ with respect to θ it comes :

$$-\|\vec{g}\|\frac{dk_{\vec{t}}}{d\theta} = \frac{d(t^T H t)}{d\theta} = \sin 2\theta(\vec{b}^T H\vec{b} - \vec{a}^T H\vec{a}) + 2\cos 2\theta \vec{a}^T H\vec{b}$$

Then if θ is a principal curvature direction this derivative is equal to 0 i.e. :

$$\frac{dk_{\vec{t}}}{d\theta} = 0 \Longleftrightarrow \tan 2\theta = \frac{2\vec{a}^T H\vec{b}}{\vec{a}^T H\vec{a} - \vec{b}^T H\vec{b}} = E \Longleftrightarrow \left(\theta = \theta_1 = \frac{\arctan E}{2}\right) \vee \left(\theta = \theta_2 = \theta_1 + \frac{\pi}{2}\right)$$

We notice that the first equivalence requires that the denominator does not vanish. In fact, the denominator is zero if and only if the point is an umbilic point [dC76]. The two principal curvature directions $\vec{t}_1$ and $\vec{t}_2$ are :

$$\vec{t}_1 = \cos\theta_1\ \vec{a} + \sin\theta_1\ \vec{b}$$

$$\vec{t}_2 = \cos\theta_2\ \vec{a} + \sin\theta_2\ \vec{b}$$

The two principal curvatures are :

$$K_1 = k_{\vec{t}_1} = \frac{\vec{t}_1^T H\vec{t}_1}{\|\vec{g}\|}$$

$$K_2 = k_{\vec{t}_2} = \frac{\vec{t}_2^T H\vec{t}_2}{\|\vec{g}\|}$$

In the sequel we shall assume that $\mid k_{\vec{t}_1} \mid > \mid k_{\vec{t}_2} \mid$ i.e. $\vec{t}_1$ is the maximum curvature direction and $k_{\vec{t}_1}$ the maximum curvature.

In order to avoid the arctangent computation the following method can also be used to find the principal curvatures and the principal curvature directions :

The principal curvatures may be characterized as the solution to the constrained optimization problem :

$$Min\ <\ H\ d\ ,\ d\ >$$

$$s.c. \begin{cases} <\ d\ ,\ d\ > = 1 \\ <\ d\ ,\ g\ > = 0 \end{cases}$$

We define a rotation matrix P to change coordinates so that the first basis vector is rotated to the direction $\vec{g}$:

$$P = \begin{pmatrix} \frac{I_x}{\delta} & \frac{I_y}{\gamma} & \frac{I_z\ I_x}{\gamma\delta} \\ \frac{I_y}{\delta} & \frac{-I_x}{\gamma} & \frac{I_y\ I_z}{\gamma\delta} \\ \frac{I_z}{\delta} & 0 & -\frac{\gamma}{\delta} \end{pmatrix} = \begin{pmatrix} \vec{g} & \vec{h} & \vec{f} \end{pmatrix}$$

with

$$\gamma = \sqrt{I_x^2 + I_y^2} \text{ and } \delta = \sqrt{I_x^2 + I_y^2 + I_z^2}$$

After the change of variables $P\ w\ =\ d$ the previous minimization problem becomes :

$$Min\ <\ H\ P\ w\ ,\ P\ w\ >$$
$$s.c. \begin{cases} <\ w\ ,\ w\ >\ =\ 1 \\ w_1\ =\ 0 \end{cases}$$

with

$$w = (w_1, w_2, w_3)^T$$

Introducing the following notations

$$H'\ =\ P^T\ H\ P\ =\ \begin{pmatrix} \cdot & \cdot\ \cdot \\ \cdot & H'_e \\ \cdot & \end{pmatrix}$$
$$w\ =\ \begin{pmatrix} \cdot \\ w_e \end{pmatrix}$$

we obtain the following equivalent problem :

$$Min \ < \ H'_e \, w_e \ , \ w_e \ >$$
$$s.c. < \ w_e \ , \ w_e \ > = \ 1$$

Using the *Lagrange multiplicators technique*, this problem can be reduced to the diagonalisation of the matrix H'_e. The eigenvalues of matrix H'_e correspond to the principal curvatures. The principal curvature directions are obtained by applying the matrix P to the eigenvectors of H'_e. This scheme allows to express principal curvatures K_i and principal curvature directions (d_i) using only first and second derivatives of the images :

$$K_i \ = \frac{h^T H h + f^T H f \ \pm \ \sqrt{(h^T H h - f^T H f)^2 \ + \ 4 \ (h^T H f)^2}}{2}$$

$$d_i \ = \begin{pmatrix} h_1 \ + \ f_1 \frac{K_i \ - \ h^T H h}{f^T H h} \\ h_2 \ + \ f_2 \frac{K_i \ - \ h^T H h}{f^T H h} \\ h_3 \ + \ f_3 \frac{K_i \ - \ h^T H h}{f^T H h} \end{pmatrix} \text{ with } i = 1, 2$$

IV Characterizing ridge points from partial derivatives

Of the two principal curvature directions at a point, one of the two has the maximum curvature. The associated direction is the maximum curvature direction. We will call a *ridge point* any point whose curvature in the maximum curvature direction is locally maximal, and a *valley point* a point where the curvature is minimum in the maximum curvature direction. That is, the maximum curvature value along integrals of the maximum curvature vector field has a local extremum at a ridge or valley point.

IV.1 2D case

It is of interest to study first the 2D case which presents no ambiguity (see figure 15).

The 2D contour can be represented by a curve $(x(s), y(s))$ where s is the arclength parameter. The curvature $k(s)$ is given by the formula previously established :

$$k(s) = \frac{-t^T H t}{\|\vec{g}\|} \text{ with } \|\vec{g}\|\vec{t} = \begin{pmatrix} -I_y \\ I_x \end{pmatrix} \text{ and } \vec{t} = \begin{pmatrix} x'(s) \\ y'(s) \end{pmatrix}$$

The curvature extrema are located where the derivative $k'(s)$ is zero i.e. :

$$\frac{\frac{d(t^T H t)}{ds}\|\vec{g}\| - (t^T H t)\frac{d\|\vec{g}\|}{ds}}{\|\vec{g}\|^2} = 0$$

Since :

$$\frac{d\|\vec{g}\|}{ds} = \frac{g^T H t}{\|\vec{g}\|}$$

$$\frac{d(t^T H t)}{ds} = \frac{dt}{ds}^T H t + t^T \frac{d(Ht)}{ds}$$

$$\frac{d(Ht)}{ds} = \begin{pmatrix} t^T H_x t \\ t^T H_y t \end{pmatrix} + H\frac{dt}{ds} \text{ with } H_x = \begin{pmatrix} I_{xxx} & I_{xyx} \\ I_{yxx} & I_{yyx} \end{pmatrix}, H_y = \begin{pmatrix} I_{xxy} & I_{xyy} \\ I_{yxy} & I_{yyy} \end{pmatrix}$$

Finally the curvature extrema are defined by the following equation :

$$\frac{3\,(t^T H t)(g^T H t) - \|\vec{g}\|^2\, t^T \begin{pmatrix} t^T H_x t \\ t^T H_y t \end{pmatrix}}{\|\vec{g}\|^3} = 0$$

IV.2 3D case

The 3D case is not exactly the extension of the 2D analysis because there is a different curvature for each surface direction, i.e. for each curve of the surface $\sum$ including a given point m and such that the curve normal corresponds to the surface normal. A local curvature extremum is a local extremum of the maximum curvature in the maximum curvature direction (the maximum curvature is the principal curvature having the highest magnitude ; the maximum curvature direction is the corresponding principal curvature direction). It is essential to remark that this definition is sound thanks to the continuity

and differentiability of the principal curvature field in a regular surface (see [dC76], page 156). Therefore we calculate the derivative of the curvature $k_{\vec{d}}$ in the direction $\vec{d}$, the curvature extrema are located where this derivative is zero in the maximum curvature direction. We notice finally that a very similar definition is presented by Yuille and Leyton [YL90] who set the curvature extremum as the principal curvature extremum along a curvature line.
Let $\vec{t_1}$ be the maximum curvature direction, the derivative of the curvature in the direction $\vec{t_1}$ is :

$$\nabla_{\vec{t_1}} k_{\vec{t_1}}(m) = \lim_{\lambda \to 0} \frac{k_{\vec{t_1}}(m + \lambda \vec{t_1}) - k_{\vec{t_1}}(m)}{\lambda} =$$

$$\nabla k_{\vec{t_1}}(m) \cdot \vec{t_1} = \begin{bmatrix} \frac{\partial k_{\vec{t_1}}}{\partial x}(m) \\ \frac{\partial k_{\vec{t_1}}}{\partial y}(m) \\ \frac{\partial k_{\vec{t_1}}}{\partial z}(m) \end{bmatrix} \cdot \begin{bmatrix} t_{1x} \\ t_{1y} \\ t_{1z} \end{bmatrix}$$

We have :

$$k_{t_1}(m) = \frac{I_{xx}t_{1x}^2 + I_{yy}t_{1y}^2 + I_{zz}t_{1z}^2 + 2I_{xy}t_{1x}t_{1y} + 2I_{xz}t_{1x}t_{1z} + 2I_{yz}t_{1y}t_{1z}}{\|\vec{g}\|^2}$$

It comes

$$\nabla_{\vec{t_1}} k_{t_1}(m) = \frac{\|\vec{g}\|^2 t_1^T \cdot \begin{bmatrix} t_1^T H_x t_1 \\ t_1^T H_y t_1 \\ t_1^T H_z t_1 \end{bmatrix} - \left(t_1^T H t_1\right)\left(t_1^T H g\right)}{\|\vec{g}\|^3}$$

with H being the hessian of I and H_x, H_y, H_z its partial derivatives.

$$H = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

$$H_x = \frac{\partial H}{\partial x} = \begin{bmatrix} I_{xxx} & I_{xyx} & I_{xzx} \\ I_{yxx} & I_{yyx} & I_{yzx} \\ I_{zxx} & I_{zyx} & I_{zzx} \end{bmatrix}$$

In the sequel we will call $\nabla_{\vec{t_1}} k_{\vec{t_1}}(m)$ the "extremality" criterion.

V Summary of the algorithms

We summarize the algorithms implied by the formulas of the previous sections that find curvatures and ridge points on surfaces in 3D data. Let $I(x, y, z)$ be a 3D image. Using a 3D edge extractor, a collection of points are designated as "surface points". Since the edge extractor may use differential information, the computations may be overlapped with portions of the first step below. The following algorithm is used to compute curvatures and ridge points directly from the image data.

1. Compute the nineteen first, second and third order partial derivatives of I. using the 3D separable recursive filters given in section 2.

2. Extract the 3D edges using either a first or a second derivative approach [MDM91]. The second derivative approach works better because the corners are also detected [DG90]. Practically, in the case where the recursive or the direct implementation by convolution masks of the filters is too prohibitive regarding the computational cost, the memory space, and the disk storage requirement, the following two stages implementation could be used : smoothing of the original data using recursive filters, derivation using small convolution masks.

3. For each edge point

   - Compute the two principal curvatures and the corresponding principal curvatures (see section 3)
   - Compute the "extremality " criterion (see section 4).
   - Using the maximum curvature and the maximum curvature direction select the edge points where the maximum curvature is locally extremum along the maximum curvature direction. We use a method described in [MAS91] very similar to the extraction of the maximum of the gradient magnitude in the gradient direction [Can86] . We can also use the extremality criterion (see section 4) to select the edge points. A point would be selected if its maximum curvature is high and if its extremality is low (we notice that the extremality has no sign). We are only starting this kind of experiments (see section 9).

VI Differential geometry in the fourth dimension space

In this section we show that differential geometry in $\mathbf{R}^4$ can be used to extract invariants from 3D images. In this case the differential characteristics (first and second fundamental forms, curvatures) depend both on the geometry of the surfaces and on the grey level function. In the sequel we simply apply some basic definitions of the differential geometry in $\mathbf{R}^4$ to 3D imagery.

Let E be the affine Euclidian oriented space of dimension 4 and V the associated vectorial space.

Definition 1 (*Parametrized hypersurface) A parametrized hypersurface of E is a couple (Σ, M) where Σ is a subset of E and M a continuous mapping from an open set U of $\mathbf{R}^3$ to E whose image $M(U)$ is equal to Σ; M is called a parametrized representation of the hypersurface Σ.*

If M is C^k the parametrized Σ hypersurface is said to be C^k.

As an application to 3D imagery :

Let I(x,y,z) be the grey level function of a 3D image and let :

$$\Sigma = \left\{\vec{v} \in \mathbf{R}^4 / \vec{v} = (x, y, z, I(x, y, z))\right\}$$

$$M : \mathbf{R}^3 \longrightarrow E$$

given by

$$(x, y, z) \longrightarrow (x, y, z, I(x, y, z))$$

Thus we can consider a 3D image as a hypersurface in $\mathbf{R}^4$.

Definition 2 (*Tangent hyperplane) Let (Σ, M) be a C^k parametrized hypersurface ($k \geq 1$). If $M_0 = M(u_0, v_0, w_0)$ is such that the vectors $\frac{\partial \vec{M}}{\partial v}, \frac{\partial \vec{M}}{\partial v}$ and $\frac{\partial \vec{M}}{\partial w}$ computed at point M_0 (this is true in the following unless specified differently) are linearly independent, then the hypersurface (Σ, M) has a tangent hyperplane at M_0 and the tangent hyperspace to Σ at M_0 is given by :*

$$T_{M_0}\Sigma = \mathbf{R}\frac{\partial \vec{M}}{\partial u} \oplus \mathbf{R}\frac{\partial \vec{M}}{\partial v} \oplus \mathbf{R}\frac{\partial \vec{M}}{\partial w}$$

The normal to the hypersurface at M_0 is given by :

$$\vec{K_0} = \frac{\partial \vec{M}}{\partial u} \wedge \frac{\partial \vec{M}}{\partial v} \wedge \frac{\partial \vec{M}}{\partial w}$$

When applied to 3D imagery we have the following :

Let $\vec{G} = (I_x, I_y, I_z)^t$ be the gradient at a point $P_0(x_0, y_0, z_0)$; (here we use subscripts to denote partial derivatives).the point $P_0 : (x_0, y_0, z_0)$ in $\mathbf{R}^3$ corresponds to the point $\vec{M}(x_0, y_0, z_0) = M_0 : (x_0, y_0, z_0, I(x_0, y_0, z_0))$ in $\mathbf{R}^4$ then :

$$\frac{\partial \vec{M}}{\partial u} = (1, 0, 0, \ I_x)^t$$

$$\frac{\partial \vec{M}}{\partial v} = (0, 1, 0 \ I_y)^t$$

$$\frac{\partial \vec{M}}{\partial w} = (0, 0, 1, \ I_z)^t$$

The normal to the hypersurface at M_0 is :

$$\vec{K_0} = \frac{\partial \vec{M}}{\partial u} \wedge \frac{\partial \vec{M}}{\partial v} \wedge \frac{\partial \vec{M}}{\partial w}$$

$$= \frac{1}{\sqrt{D}}(I_x, I_y, I_z, -1)^t$$

with :

$$D = 1 + I_x^2 + I_y^2 + I_z^2 = \|\nabla M\|^2$$

Definition 3 (*First fundamental form) The restriction of the euclidian scalar product to the tangent space $T_{M_0}\Sigma$ to Σ at M_0 is an euclidian scalar product on this vector space which we denote $\phi_1(M_0) = \phi_1(u_0, v_0, w_0)$. Given that $(\frac{\partial \vec{M}}{\partial u}\frac{\partial \vec{M}}{\partial v}, \frac{\partial \vec{M}}{\partial w})$ is a basis of $T_{M_0}\Sigma$ the matrix of ϕ_1 for this basis is :*

$$F_1 = \begin{pmatrix} \frac{\partial \vec{M}}{\partial u} \cdot \frac{\partial \vec{M}}{\partial u} & \frac{\partial \vec{M}}{\partial u} \cdot \frac{\partial \vec{M}}{\partial v} & \frac{\partial \vec{M}}{\partial u} \cdot \frac{\partial \vec{M}}{\partial w} \\ \frac{\partial \vec{M}}{\partial v} \cdot \frac{\partial \vec{M}}{\partial u} & \frac{\partial \vec{M}}{\partial v} \cdot \frac{\partial \vec{M}}{\partial v} & \frac{\partial \vec{M}}{\partial v} \cdot \frac{\partial \vec{M}}{\partial w} \\ \frac{\partial \vec{M}}{\partial w} \cdot \frac{\partial \vec{M}}{\partial u} & \frac{\partial \vec{M}}{\partial w} \cdot \frac{\partial \vec{M}}{\partial v} & \frac{\partial \vec{M}}{\partial w} \cdot \frac{\partial \vec{M}}{\partial w} \end{pmatrix} = \begin{pmatrix} e & f & h \\ f & g & i \\ h & i & j \end{pmatrix}$$

ϕ_1 is called the first fundamental form of the hypersrface Σ at M_0.

A direct application of the definition 3 gives the first fundamental from of the hypersurface defined by the grey level function at a point P_0 :

$$F_1 = \begin{pmatrix} 1+I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & 1+I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & 1+I_z^2 \end{pmatrix}$$

Definition 4 (*second fundamental form*) *The second fundamental form of the hypersurface at a point M_0 is defined by a matrix in the basis ($\frac{\partial \vec{M}}{\partial u}$, $\frac{\partial \vec{M}}{\partial v}(u_0,v_0,w_0)\frac{\partial \vec{M}}{\partial w}(u_0,v_0,w_0)$), using the normal to the surface $K=(u_0,v_0,w_0)$:*

$$F_2 = \begin{pmatrix} l & m & o \\ m & n & p \\ o & p & q \end{pmatrix}$$

with :

$$l = -\frac{\partial \vec{M}}{\partial u}\cdot\frac{\partial \vec{K}}{\partial u} = \vec{K}\cdot\frac{\partial \vec{M}}{\partial u^2}$$

$$m = -\frac{\partial \vec{M}}{\partial u}\cdot\frac{\partial \vec{K}}{\partial v} = \vec{K}\cdot\frac{\partial^2 \vec{M}}{\partial u\partial v}$$

$$o = -\frac{\partial \vec{M}}{\partial u}\cdot\frac{\partial \vec{K}}{\partial w} = \vec{K}\cdot\frac{\partial^2 \vec{M}}{\partial u\partial w}$$

$$p = -\frac{\partial \vec{M}}{\partial v}\cdot\frac{\partial \vec{K}}{\partial w} = \vec{K}\cdot\frac{\partial^2 \vec{M}}{\partial v\partial w}$$

$$q = -\frac{\partial \vec{M}}{\partial w}\cdot\frac{\partial \vec{K}}{\partial w} = \vec{K}\cdot\frac{\partial^2 \vec{M}}{\partial w^2}$$

$$n = -\frac{\partial \vec{M}}{\partial v}\cdot\frac{\partial \vec{K}}{\partial v} = \vec{K}\cdot\frac{\partial^2 \vec{M}}{\partial v^2}$$

A direct application of the definition 4 gives the second fundamental form at a point $P_0 = (x_0, y_0, z_0)$

$$F_2 = \frac{1}{\sqrt{D}}\begin{pmatrix} -I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & -I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & -I_{zz} \end{pmatrix}$$

Definition 5 *The Weingarten endomorphism of the hypersurface at the point M_0 is the mapping L_0 such that :*

$$L_0 : T_{M_0}\Sigma \longrightarrow T_{M_0}\Sigma$$

$$L_0(X_u\frac{\partial \vec{M}}{\partial u} + X_v\frac{\partial \vec{M}}{\partial v} + X_w\frac{\partial \vec{M}}{\partial w})$$

$$= -X_u\frac{\partial \vec{K}}{\partial u} - X_v\frac{\partial \vec{K}}{\partial v} - X_w\frac{\partial \vec{K}}{\partial w}$$

It can easily be shown that the matrix of L_0 in the basis $(\frac{\partial \vec{M}}{\partial u}, \frac{\partial \vec{M}}{\partial v}, \frac{\partial \vec{M}}{\partial w})$ is given by :

$$W = F_2F_1^{-1}$$

where F_1 and F_2 are respectively the matrix of the first and second fundamental form in the basis $(\frac{\partial \vec{M}}{\partial u}\frac{\partial \vec{M}}{\partial v}, \frac{\partial \vec{M}}{\partial w})$.

Definition 6 *The eigenvectors of L_0 which we denote $\vec{X}_1, \vec{X}_2, \vec{X}_3$ are called the principal curvature directions and the corresponding eigenvalues : $\lambda_1, \lambda_2, \lambda_3$ are called the principal curvatures of (Σ, M) at M_0. The product $\lambda_1\lambda_2\lambda_3 (= \det W)$ is called the gaussian curvature. The mean of the curvature values $\frac{\lambda_1 + \lambda_2 + \lambda_3}{3} (= \frac{trace\, W}{3})$ is called the mean curvature*

Remark 1 *In this case the sign of the gaussian curvature is not invariant [Spi71] with surface orientation.*

Definition 7 *The eingenvector of L_0 corresponding to the eigenvalue whose magnitude is maximum is called the maximum curvature direction $\vec{X}_m$ and the value of the eigenvalue λ_m is called the maximum curvature.*

Remark 2 *Our experiments seem to show that maximum curvature and maximum curvature direction are the most reliable differential characteristics.*

The application of the definitions 5,6,7 to the 3D image case are straightforward

VII Relationships between the curvatures of the hypersurface and the curvatures of the surface

In this section we show that the maximum curvature of the surface can be recovered using the maximum curvature of the hypersurface. We use a 3D roof as surface model and a 3D version of Deriche filter [MDM91] to compute the first and second order derivatives.

VII.1 Operators to calculate first and second derivative of the 3D image

We set

$$\begin{aligned} l(x) &= \frac{\alpha}{4}(\alpha \mid x \mid +1)\, e^{-\alpha|x|} && \text{(smoothing filter)} \\ d(x) &= \alpha^2 x\, e^{-\alpha|x|} && \text{(first derivative filter)} \\ s(x) &= (1-\alpha \mid x \mid)\, e^{-\alpha|x|} && \text{(second derivative filter)} \end{aligned}$$

We notice that these filters are normalized in the **continuous** domain for our calculus, but in the implementation the normalization should be done in the discrete domain. For $I(x,y,z)$ a 3D image we compute the first and second partial derivatives as a convolution product (see section II).

VII.2 Geometric model

We consider a 3D infinite roof step edge $R(x,y,z)$ (see figure VII.2) defined as follows :

$$R(x,y,z) = \begin{cases} A & \text{if } (y \geq 0) \wedge (-y \leq kx \leq y) \\ 0 & \text{otherwise} \end{cases}$$

VII.3 Calculation of the first and second partial derivatives at the origin

Using the assistance of the symbolic program maple we obtain the following results for the roof edge :

$$I_x(0,0,0) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} R(x,y,z)d(x)l(y)l(z)dxdydz = 0$$

$$I_y(0,0,0) = \int_{\infty}^{+\infty}\int_{\infty}^{+\infty}\int_{\infty}^{+\infty} R(x,y,z)l(x)d(y)l(z)dxdydz = \frac{A(1+3k+k^2)}{(1+k)^3}$$

$$I_x(0,0,0) = I_{xy}(0,0,0) = I_{xz}(0,0,0) = I_{yz}(0,0,0) = I_{zz}(0,0,0) = 0$$

$$I_{xx}(0,0,0) = \frac{Ak(1+3k)}{2\alpha(1+k)^3}$$
$$I_{yy}(0,0,0) = \frac{-Ak(k+3)}{2\alpha(1+k)^3}$$

VII.4 Calculation of the curvatures of the hypersurface

The first fundamental form at the origin is :

$$F1 = \begin{bmatrix} 1+I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & 1+I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & 1+I_z^2 \end{bmatrix} = \begin{bmatrix} 1+I_x^2 & I_xI_y & 0 \\ I_xI_y & 1+I_y^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

with :

$$\begin{cases} 1+I_x^2 = 1 \\ 1+I_y^2 = 1 + A^2\dfrac{(1+3k+k^2)^2}{(1+k)^6} \\ I_xI_y = 0 \end{cases}$$

The second fundamental form at point 0 is :

$$F_2 = \frac{1}{-\sqrt{D}}\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{xz} & I_{zz} \end{bmatrix} = -\frac{1}{\sqrt{D}}\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

with :

$$I_{xx} = \frac{A\,k(1+3k)}{2\alpha(1+k)^3}$$
$$I_{yy} = -\frac{A\,k(k+3)}{2\alpha(1+k)^3}$$
$$D = 1 + I_x^2 + I_y^2 + I_z^2 = 1 + I_y^2 = 1 + A^2\frac{(1+3k+k^2)^2}{(1+k)^6}$$

The Weingarten endomorphism at point 0 is :

$$W = F_2 F^{-1} = \begin{bmatrix} \frac{I_{xx}}{(1+I_y^2)^{1/2}} & 0 & 0 \\ 0 & \frac{I_{yy}}{(1+I_y^2)^{3/2}} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Setting C_1, C_2, C_3 to the principal curvatures of the hypersurface at the point 0 we see that :

$$\begin{aligned} C_1 &= \frac{I_{xx}}{(1+I_y^2)^{1/2}} \\ C_2 &= \frac{I_{yy}}{(1+I_y^2)^{3/2}} \\ C_3 &= 0 \end{aligned}$$

and that $\vec{P}_1, \vec{P}_2, \vec{P}_3$ the principal curvature directions at point 0 of the hypersurface are :

$$\vec{P}_1 = (1,0,0)$$
$$\vec{P}_2 = (0,1,0)$$
$$\vec{P}_3 = (0,0,1)$$

in the basis $(\frac{\partial \vec{M}}{\partial u}, \frac{\partial \vec{M}}{\partial v}, \frac{\partial \vec{M}}{\partial w})$ we have :

$$\mid \frac{C_1}{C_2} \mid = \mid \frac{I_{xx}}{I_{yy}} \mid (1+I_y^2) = (\frac{1+3k}{k+3})(1+A^2\frac{(1+3k+k^2)^2}{(k+1)^6})$$

the minimum value of $\mid \frac{C_1}{C_2} \mid$ is obtained for $k = 0$ *i.e.* $\frac{1}{3} + \frac{A^2}{3}$ thus

$$A > \sqrt{2} \Longrightarrow \mid \frac{C_1}{C_2} \mid > 1$$

Moreover we notice that

$$k > 1 \Longrightarrow \mid \frac{I_{xx}}{I_{yy}} \mid > 1 \Longrightarrow \mid \frac{C_1}{C_2} \mid > 1$$

We notice that $k = 1$ corresponds to an angle of $\frac{\Pi}{2}$ which is equivalent to the curvature of the step edge. However in the real case the value of A is always greater than $\sqrt{2}$ thus we can suppose :
$\mid \frac{C_1}{C_2} \mid > 1 \Longleftrightarrow C_1$ is the maximum curvature of the hypersurface.

We obtain the remarkable result that the maximum curvature at point 0 of the hypersurface is :

(6) $$C_1 = \frac{Ak(1+3k)}{2\alpha(1+k)^3} \frac{1}{\sqrt{1+\frac{A^2(1+3k+k^2)^2}{(k+1)^6}}}$$

and that the corresponding maximum curvature direction is :

$$\vec{P_1} = \frac{\partial \vec{M}}{\partial u}(x_0, y_0, z_0) = (1,0,0,0)^t$$

in the basis (X, Y, Z, T)

VII.5 Linking surface and hypersurface maximum curvatures

Up to now we have considered our 3D roof step edge as an hypersurface in R^4 and we have calculated its maximum curvature and the corresponding maximum curvature direction. We can also consider our 3D roof step edge as a surface in R^3 determined by a threshold between 0 and A and compute the maximum curvature and the principal direction at the origin. In reality the image data must first be smoothed some and therefore the resulting curvatures will be parametrized.

The maximum curvature direction of the surface at point 0 is $(1,0,0)^t = \vec{A}$

The maximum curvature at point 0 is : $p = 2k$.
We notice that the surface is not derivable at point 0, but the curvature in the direction $0x$ is the second derivative of the section of R with $0xy$ which is approximated by the difference between the first derivatives on the two sides i.e. $k - (-k) = 2k$. Therefore using relation 6 we obtain :

(7) $$C_1(p) = \frac{Ap}{4\alpha} \frac{(1+\frac{3}{2}p)}{(1+\frac{p}{2})^3} \frac{1}{\sqrt{\frac{1+A^2(1+\frac{3}{2}p+\frac{p^2}{4})^2}{(\frac{p}{2}+1)^6}}} \quad (5)$$

Figure 4 shows the graph of $C_1(k)$. We notice that the derivative of $C_1(k)$ is zero for $C \simeq 110 (k = 55)$ which corresponds to an angle of two degrees. Such sharp angles are degenerated cases because the filters cannot detect edges on such corners. Therefore the graph of figure 4 provides the correspondance between the maximum curvature of the hypersurface and the maximum curvature of the surface : **an increasing monotonic function for practical values of A and k.**

In the previous development we have shown that in the basis (X, Y, Z, T) the maximum curvature direction of the hypersurface is : $\vec{P_1} = (1,0,0,0)^t$. In the basis (X, Y, Z) the maximum curvature direction of the surface is

$\vec{A} = (1,0,0)^t$. Therefore the first three components of the maximum curvature direction of the hypersurface give the maximum curvature direction of the surface. We notice that for filters wich are isotrope the computations we have done remain still exactly valid for any frame axis. The only non trivial filter which is isotrope is the gaussian, but Deriche filter is " almost" frame-invariant [MDM91]. Anyway we could do the same calculations for Gaussian filters but the result will not be check so clearly because of the non integrability of the Gaussian.

VII.6 Using a more complex geometric model

The same calculation could be done using a more complex geometric model having two principal curvatures not equal to zero (see figure VII.6). The relationships between the two maximum 4D curvature and the two principal 3D curvatures are also monotonic for pratical cases (see figures 6 and 7).

The computation scheme is the same as in the last section. Partial derivatives are computed at the oigin :

$$R(x,y,z) = \left\{ \ (x,y,z)/(y \geq 0) \wedge (-k_1 y \leq x \leq k_2 y) \wedge (-k_2 y \leq z \leq k_2 y) \ \right\}.$$

$$I_y = A\left[1 - \frac{(1+2k_2)}{(1+k_2)^3} - \frac{(1+2k_1)}{(1+k_1)^3} + \frac{2+6(k_1+k_2)+4(k_1^2+k_2^2)+11k_1k_2}{2(1+k_1+k_2)^4}\right]$$

$$I_{yy} = \frac{A}{2\alpha}\left[\frac{-k_2(1+3k_2)}{(1+k_2)^3} - \frac{k_1(1+3k_1)}{(1+k_1)^3} + \frac{k_1+k_2+4(k_1^2+k_2^2)+3(k_1^3+k_2^3)}{(1+k_1+k_2)^4} + 2k_1k_2(3+5(k_1+k_2))\right]$$

$$I_{xx} = \frac{Ak_1}{2\alpha}\left[\frac{3+k_1}{(1+k_1)^3} - \frac{3+4k_1+8k_2+k_1^2+3k_1k_2+2k_2^2}{(1+k_1+k_2)^4}\right]$$

$$I_{zz} = \frac{Ak_2}{2\alpha}\left[\frac{3+k_2}{(1+k_2)^3} - \frac{3+4k_2+8k_1+k_2^2+3k_1k_2+2k_2^2}{(1+k_1+k_2)^4}\right]$$

$$I_x = I_z = I_{xz} = I_{yz} = I_{xy} = 0$$

The first fundamental form at the origin is :

$$F1 = \begin{bmatrix} 1+I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & 1+I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & 1+I_z^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & D & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

with $D = 1 + I_y^2$.

$$F_2 = \frac{1}{-\sqrt{D}} \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{xz} & I_{zz} \end{bmatrix} = -\frac{1}{\sqrt{D}} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

The Weingarten endomorphism is :

$$W = F_2 F^{-1} = \begin{pmatrix} \frac{I_{xx}}{D^{1/2}} & 0 & 0 \\ 0 & \frac{I_{yy}}{D^{3/2}} & 0 \\ 0 & 0 & \frac{I_{zz}}{D^{1/2}} \end{pmatrix}$$

It comes that the three principal curvatures are :

$$\begin{aligned} C_1 &= \frac{I_{xx}}{(1 + I_y^2)^{1/2}} \\ C_2 &= \frac{I_{yy}}{(1 + I_y^2)^{3/2}} \\ C_3 &= \frac{I_{zz}}{(1 + I_y^2)^{1/2}} \end{aligned}$$

Using a similar scheme than in the previous section we can show that the two curvatures having the highest magnitude are C_1 and C_3. Looking at the graphs of $C_1(k_1, k_2)$ and $C_3(k_1, k_2)$ we also see that for practical cases $C_1(k_1, k_2)$ and $C_3(k_1, k_2)$ are increasing monotonic functions of the two principal curvatures of the surface (see figure 6).

VIII Algorithm to extract ridge lines in a 3D image using a hypersurface

Let $I(x, y, z)$ be a 3D image the following practical algorithm derives directly from the preceeding sections :

a) computation of $I_x, I_y, I_z, I_{xy}, I_{xz}I_{yz}, I_{zz}, I_{yy}$ using the 3D separable recursive filters recalled in the preceeding section.

b) extraction of the 3D edges using first or second derivative approach [MDM91].

c) For each edge point :

- compute the first and second fundamental forms of the hypersurface :

$$F_1 = \begin{bmatrix} 1+I_x^2 & I_xI_y & I_xI_z \\ I_xI_y & 1+I_y^2 & I_yI_z \\ I_xI_z & I_yI_z & 1+I_z^2 \end{bmatrix}$$

$$F_2 = \frac{1}{\sqrt{1+I_x^2+I_y^2+I_z^2}} \begin{pmatrix} -I_{xx} & -I_{x_y} & -I_{xz} \\ -I_{xy} & -I_{y_y} & -I_{yz} \\ -I_{xz} & -I_{y_z} & -I_{zz} \end{pmatrix}$$

- compute the weingarten endomorphism

$$W = F_2 F_1^{-1}$$

- Compute the eigenvectors and the eigenvalues of Weingarten endomorphism. Find the eigenvector $\vec{P_1} = x_p\frac{\partial \vec{M}}{\partial u} + y_p\frac{\partial \vec{M}}{\partial v} + z_p\frac{\partial \vec{M}}{\partial v} + j_p\frac{\partial \vec{M}}{\partial w}$ corresponding to the eigenvalue having the maximum magnitude (C_1). C_1 is the maximum curvature of the hypersurface and $\vec{P_1}$ the maximum curvature direction.

- compute the maximum curvature of the surface (p) using the graph of $C_1(p)$ (figure 4). (Practically the maximum curvature of the hypersurface is a good approximation of the maximum curvature of the surface).

- compute $\vec{A}$ the maximum curvature direction of the surface using $\vec{P_1}$.

$$\vec{P_1} = x_p\frac{\partial \vec{M}}{\partial u} + y_p\frac{\partial \vec{M}}{\partial v} + z_p\frac{\partial \vec{M}}{\partial w} =$$

$$x_p(1,0,0,I_x)^t + y_p(0,1,0,I_y) + z_p(0,0,1,I_z)$$

$$= (x_p, y_p, z_p, x_pI_x + y_pI_y + z_pI_z)$$

$$\Longrightarrow \vec{A} = (x_p, y_p, z_p)^t \text{(see the previous section)}$$

- Using the maximum curvature and the maximum curvature direction at each edge point we select the edge points where the maximum curvature is locally extremum along the maximum curvature direction. We use a method described in [MAS91] very similar to the extraction of the maximum of the gradient magnitude in the gradient direction described in [Can86].

IX Experimental results

We have tested our two methods on X-ray scanner images of the skull where we extract the maximum curvatures and the ridge lines (see figures 8 to 16 for the direct estimation, see figures 17 to 21 for the hypersurface based algorithm). The results are very similar to the ones provided by the approximation method described in [MAS91] . We notice the good continuity of the values of the maximum curvature due to the coherence of the approach : filtering and curvature calculations are merged. We also point out the low computational cost of this method compared to a classical approximation algorithm (only 3 complementary scanning of the image with a recursive filter are necessary after the 3D edge detection). In order to check the stability of our results we have tested our approach on two different images of a same skull (position A and position B) using the rigid transformation from A to B [GA91]. Our experiments show the invariance of the position of the ridge points.

X Conclusion

We have presented a new approach to extract differential characteristics of surfaces in 3D images from the second partial derivatives of the grey level function provided by separable recursive filters [MDM91, MDR91]. We have developped two approaches one working in $\mathbb{R}^3$ and the other working in $\mathbb{R}^4$.

It yields a very compact and efficient approach where filtering is used for both edge detection and curvature extraction. We obtain interesting experimental results on real data that can be finally compared with more classical approaches [SZ90, BPYA85, PB85]. Moreover the computational cost of our algorithm is very low (some filtering more than edge detection).

We point out that smoothing in the four dimensional space could be more efficient than smoothing in the three dimensional space. For instance, in the case where the contours are not iso-contours (i.e. the gradient at an edge point does not approximate the normal to the surface) the only differential invariants of the image are in $\mathbb{R}^4$. However for the images presented here the results are very similar [MBF92]. We stress that to detect typical 3D edge points such as corners and vertices an approach working in $\mathbb{R}^3$ could not be used. The only way to caracterize these points is to consider them as singularities of the hypersurface. Similarly for 2D images one should consider the image as a surface to detect vertices and corners [Nob88, DG90].

Acknowledgements

This work was partially supported by Digital Equipment Corporation.

Special thanks are due to Olivier Faugeras who did the direct calculus of the curvatures from the partial derivatives of the grey level function on the corner of a table during an ORASIS meeting in Nancy and to Bob Hummel for a substantial help in the redaction of this paper. We thank Gregoire Malandain and Rachid Deriche for some suggestions on the calculation of the partial derivatives. We thank Nicholas Ayache for some suggestions during this work and Jacques Demongeot for his encouragements.

Aubertine Raboteur provided a substantial help in the preparation of this manuscript.

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.

References

[BPYA85] Michael Brady, Jean Ponce, Alan Yuille, and Haruo Asada. Describing surfaces. In Hideo Hanafusa and Hirochika Inoue, editors, *Proceedings of the Second International Symposium on Robotics Research*, pages 5–16, Cambridge, Mass., 1985. MIT Press.

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

[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.

[Der89] R. Deriche. Fast algorithms for low level vision. *IEEE Transactions on Pattern Analysis and machine Intelligence*, 1989.

[DG90] Rachid Deriche and Gerard Giraudon. Accurate corner detection : An analytical study. *International Conference in Computer Vision, Osaka, Japan*, 1990.

[GA91] A. Gueziec and N. Ayache. Lissage et reconnaissance de courbes gauches bruitees. In *Congres AFCET*, Lyon, 1991.

[KB76] Y. Kerbrat and J. M. Braemer. *Geometrie des courbes et des surfaces.* Hermann, Paris, collection methodes, 1976.

[Kob] Nomizu Kobayashi. *Fundations of Differential Geometry.* John Wiley.

[Koe90] Jan J. Koenderink. *Solid Shape.* MIT Press, Boston, 1990.

[MAS91] Olivier Monga, Nicholas Ayache, and Peter Sander. From voxel to curvature. In *IEEE Conference on Vision and Pattern Recognition*, Hawaii, June 1991.

[MBF92] Olivier Monga, Serge Benayoun, and Olivier D. Faugeras. Using third order derivatives to extract ridge lines in 3d images. In *submitted to IEEE Conference on Vision and Pattern Recognition*, Urbana Champaign, June 1992.

[MDM91] Olivier Monga, Rachid Deriche, and Gregoire Malandain. Recursive filtering and edge closing: two primary tools for 3d edge detection. *Image and Vision Computing, Vol. 9, Number 4*, August 1991. A shortened version is in proc. of ECCV'90, Lecture notes in Computer Science 427.

[MDR91] Olivier Monga, Rachid Deriche, and Jean-Marie Rocchisani. 3d edge detection using recursive filtering: Application to scanner images. *Computer Vision Graphic and Image Processing, Vol. 53, No 1, pp. 76-87*, January 1991.

[Nob88] J.A. Noble. Finding corners. *Image and Vision Computing*, 6(9):121–128, May 1988.

[PB85] Jean Ponce and Michael Brady. Toward a surface primal sketch. In *Proceedings, IJCAI*, 1985.

[Spi71] M. Spivak. *A comprehensive introduction to differential geometry*, volume 1 to 5. Berkley, 1971.

[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.

[VF92] Thierry Viewille and Olivier Faugeras. Robust and fast computation of high order spatial derivatives in images sequences. *INRIA Report research*, 1992.

[YL90] A. Yuille and M. Leyton. 3d symmetry-curvature duality theorems. *Computer Vision, Graphics, and Image Processing*, 52:124–140, 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.

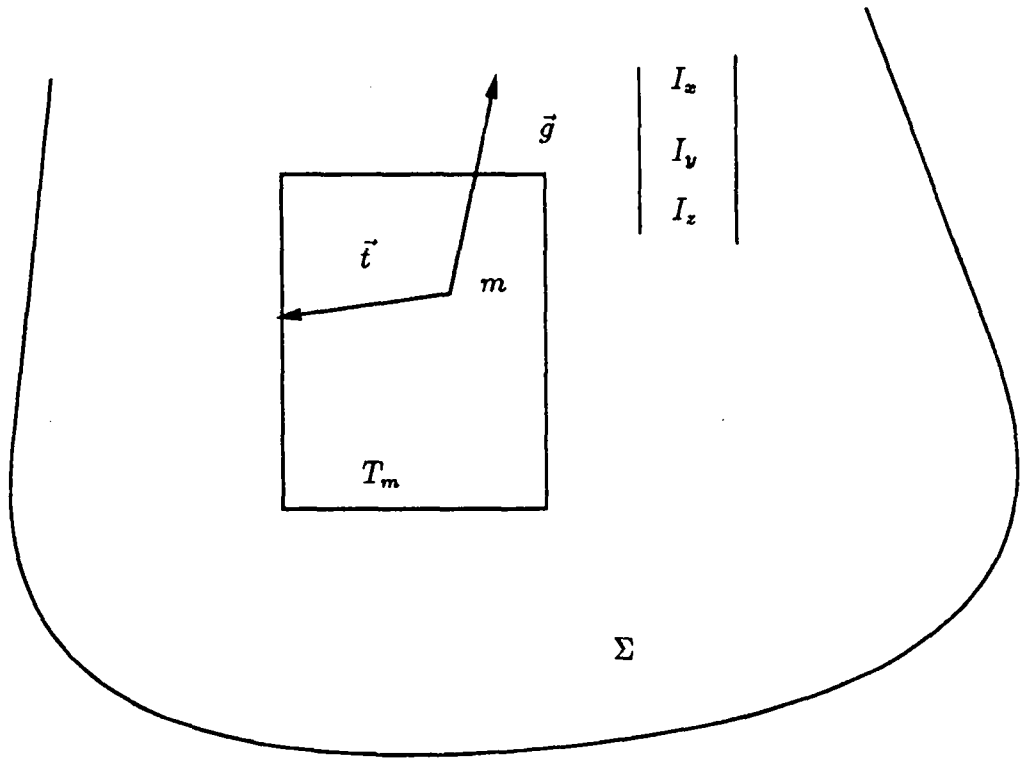


Figure 2:

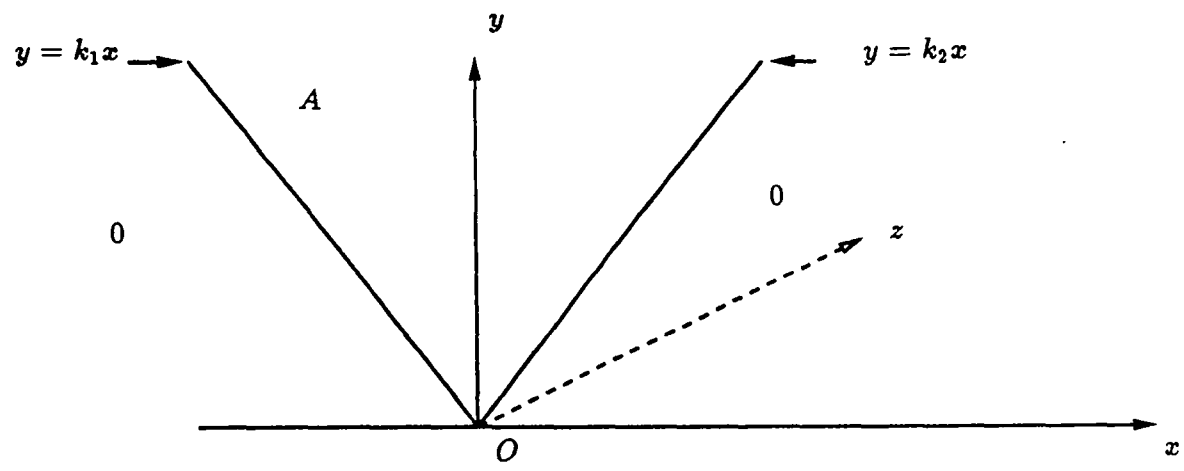


Figure 3: Section of the 3D roof step edge in Oxy plane ($k_1 = k_2 = k$, $k > 0$)

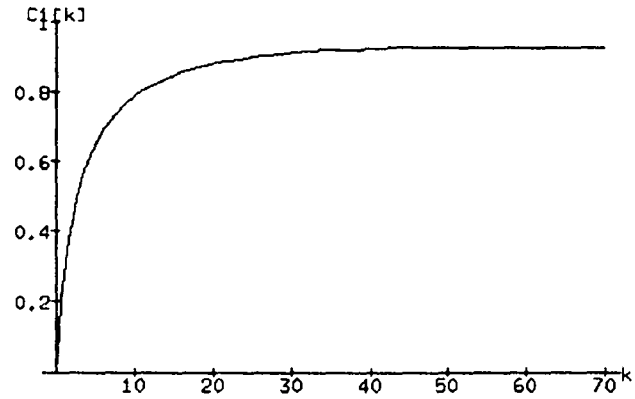


Figure 4: Graph of $C1(k)$

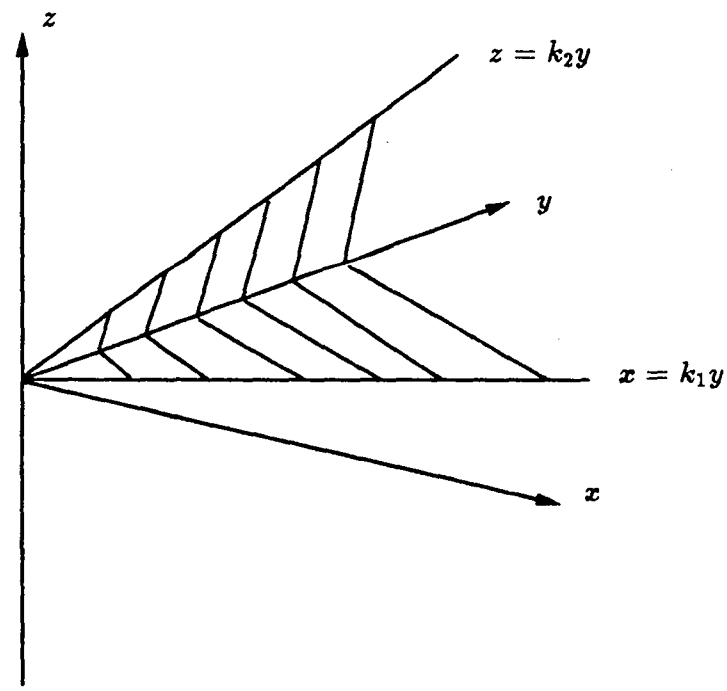


Figure 5: Section of the generalized 3D roof step edge in Oxy plane ($k > 0$)

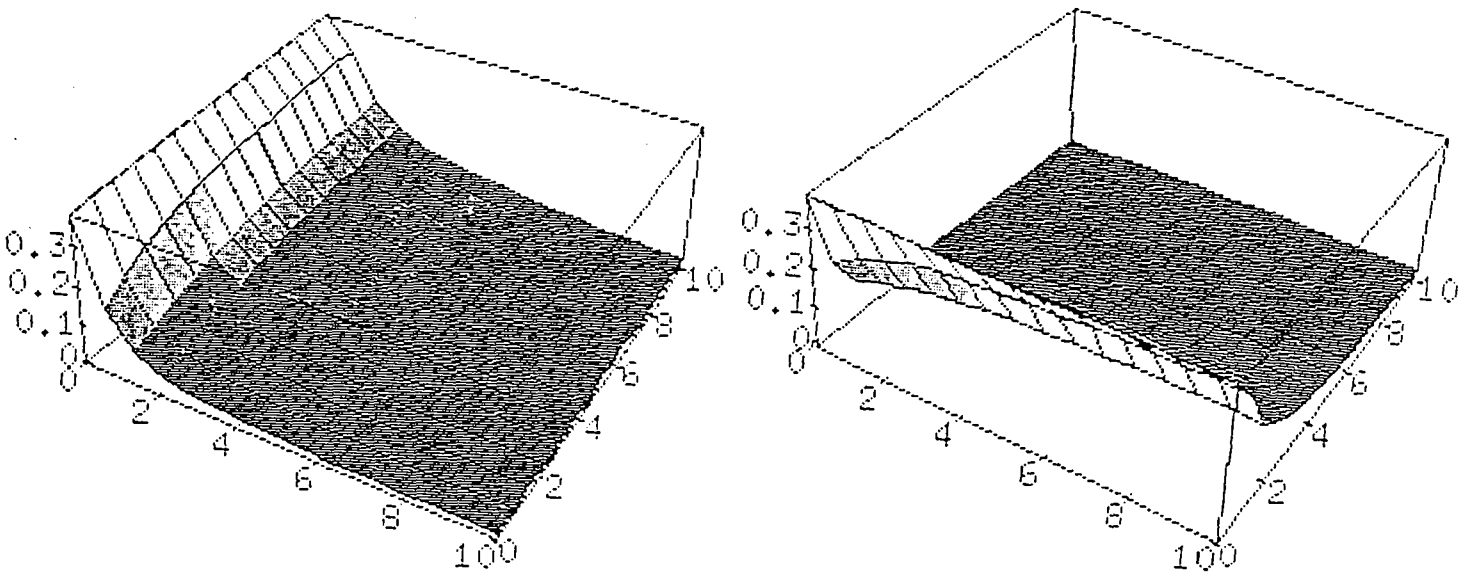


Figure 6: On left, Graph of $C1(k_1, k_2)$ and on right, Graph of $C3(k_1, k_2)$

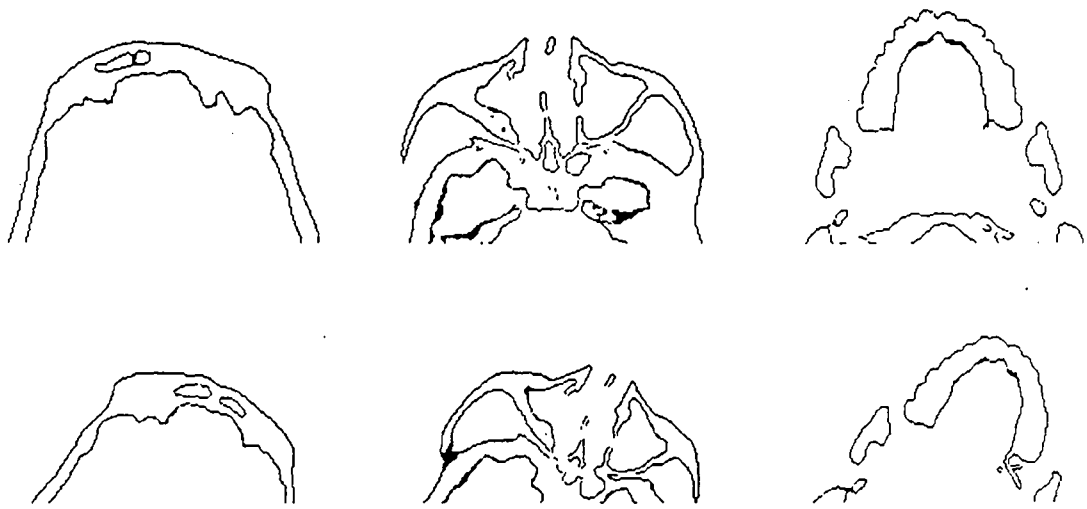


Figure 7: Cross sections of the 3D edges of two X-scanner images of the same skull with two different positions A and B (sizes of the images : 192.128.151 for the position A and 220.128.148 for the position B). Up : cross sections corresponding to position A, bottom : cross sections corresponding to position B.

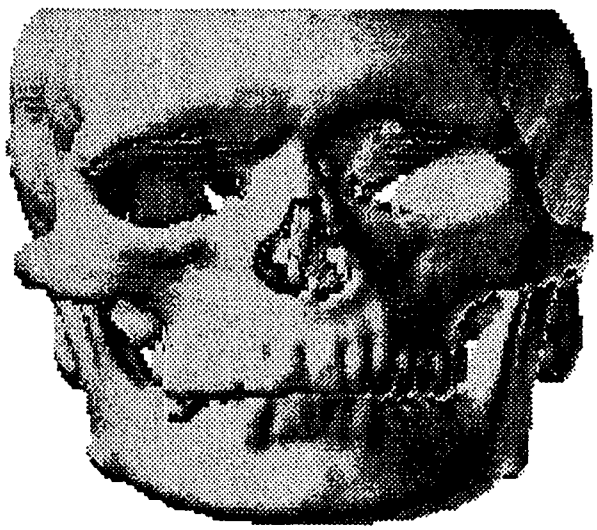


Figure 8: Perspective view of the result of the 3D edge detection for a skull.

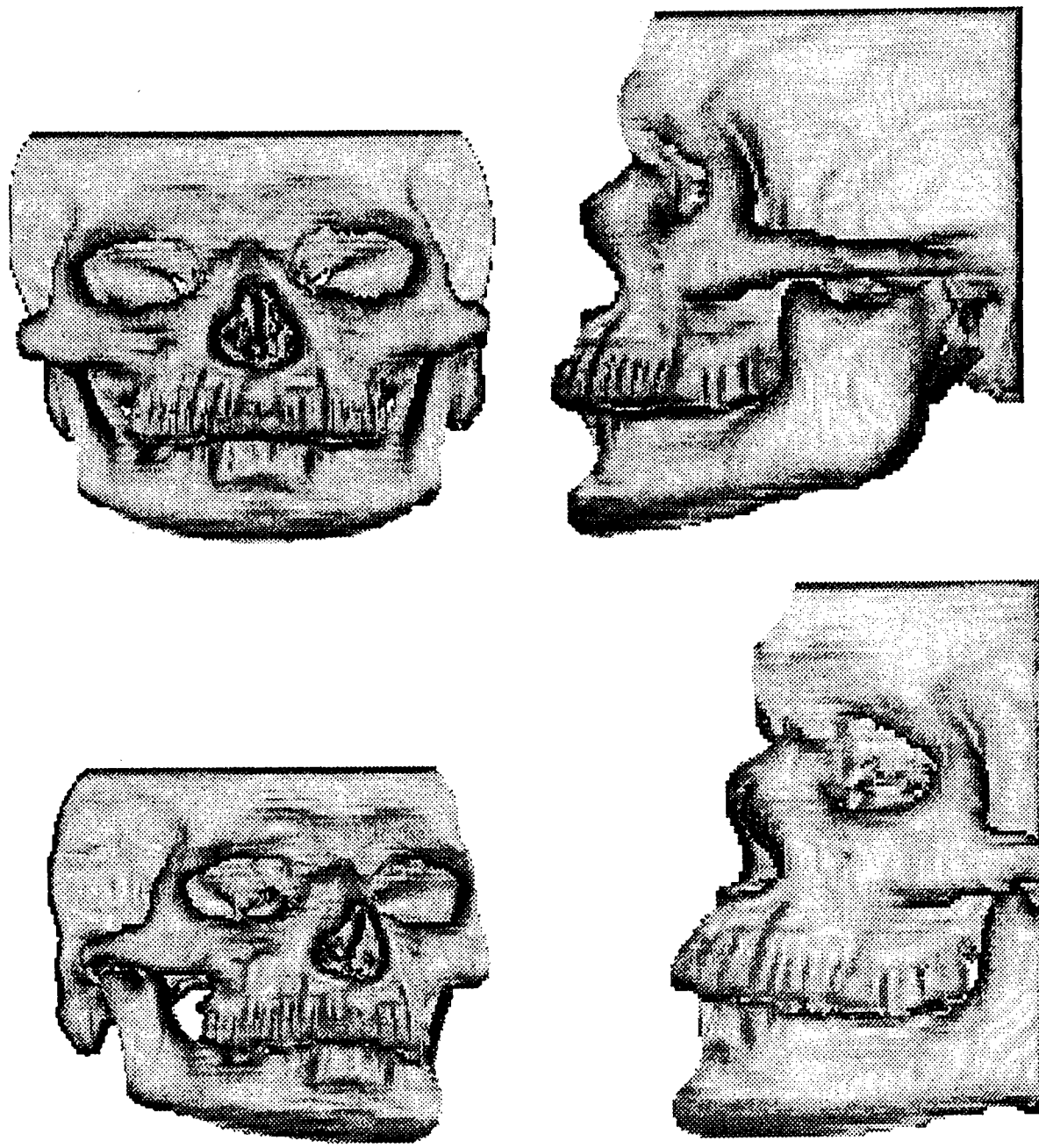


Figure 9: Perspective views of the maximum curvatures for the two positions, no rendering is done : the magnitude of the maximum curvature is displayed (up left : face view for position A, up right : profile view for position A, bottom left : face view for position B, bottom right : profile view for position B). The parameters of the filters are : $\alpha = 1.0$ for the first derivatives, $\alpha = 0.7$ for the second derivatives

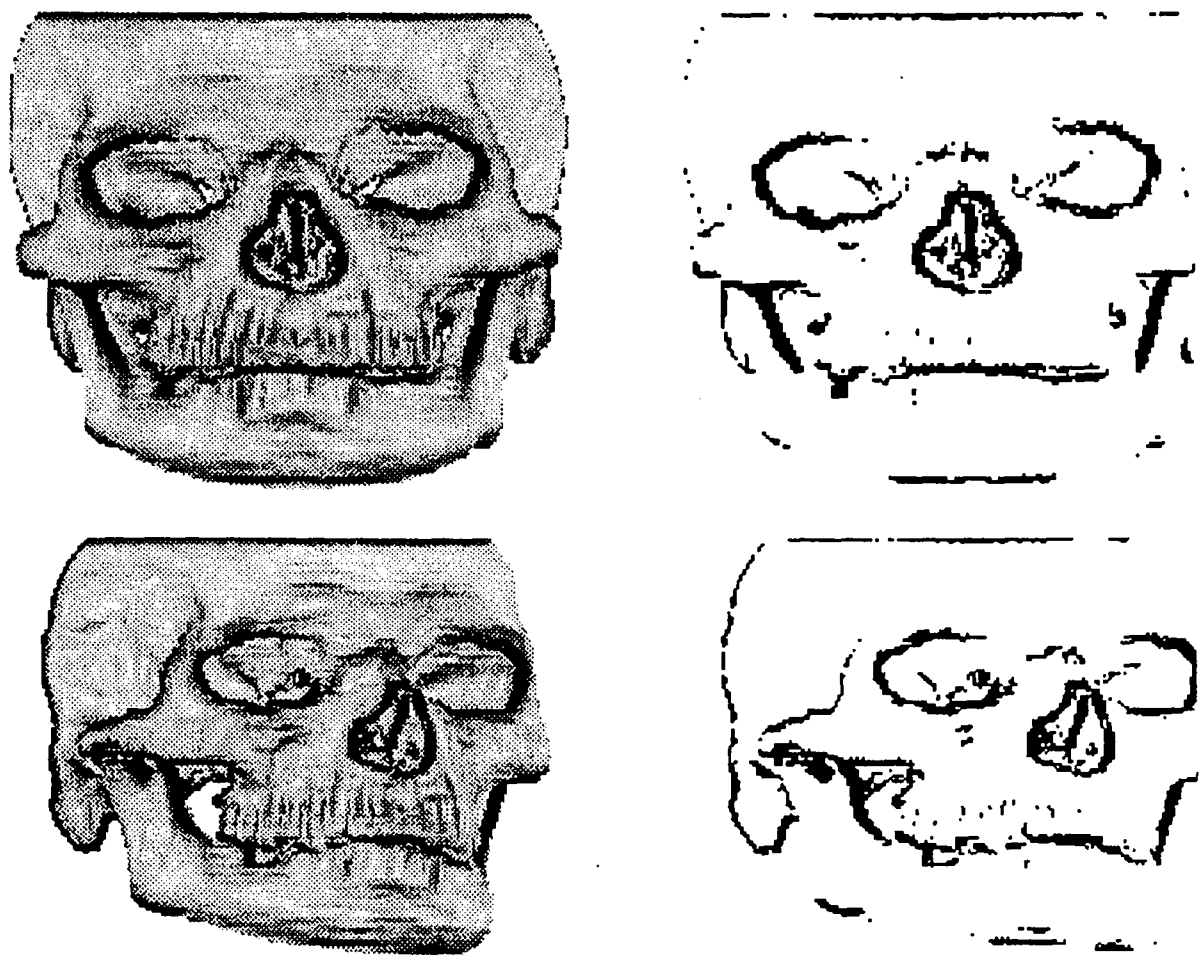


Figure 10: Images obtained by thresholding the maximum curvature. Up : face view for the position A (the high maximum curvature points are in dark) ; bottom : face view for the position B

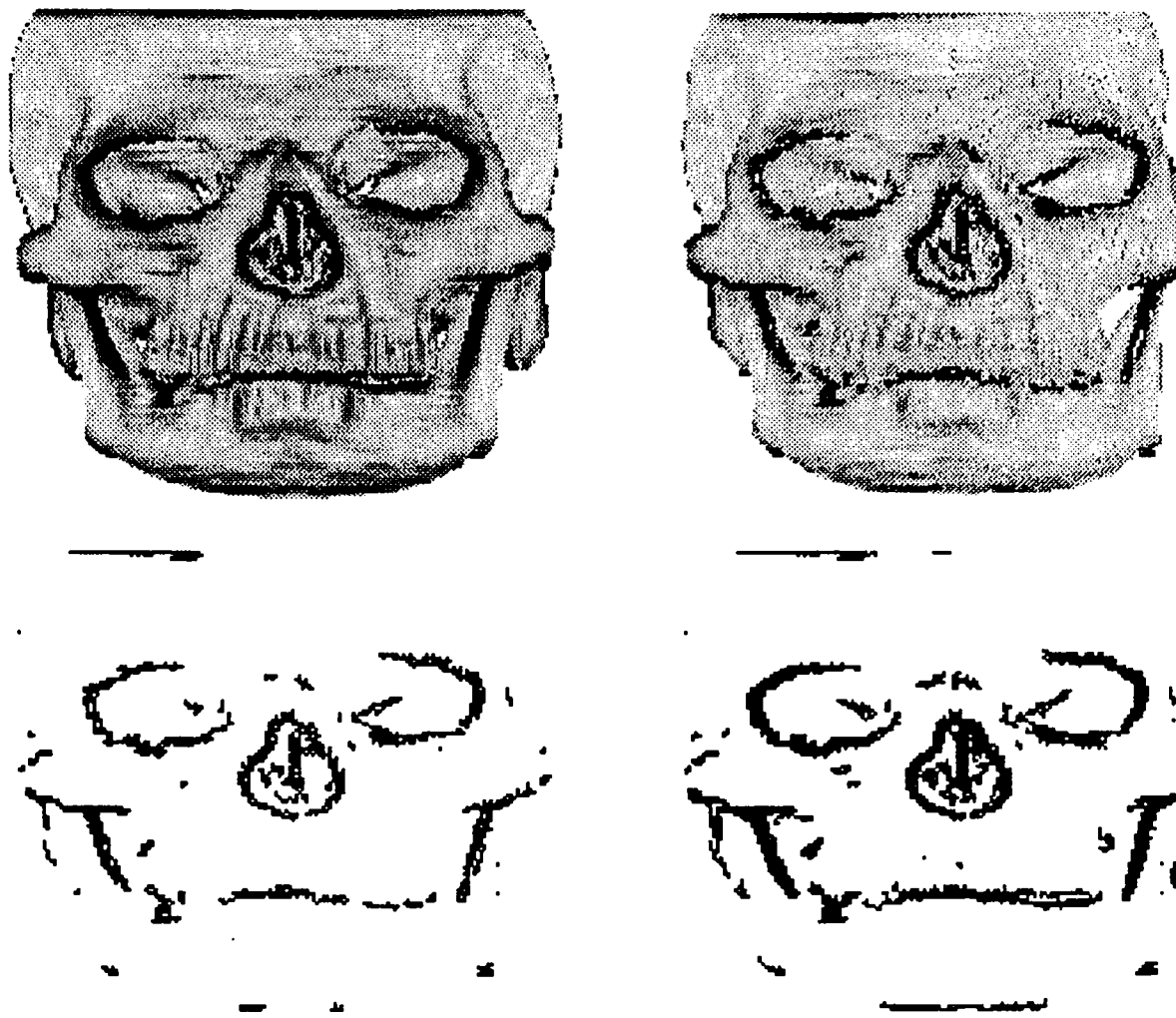


Figure 11: Up left : thresholding of the maximum curvatures for the position A, Up right : same for the position B after applying the rigid transformation (rotation and translation) from B to A, bottom right : superposition of the two results

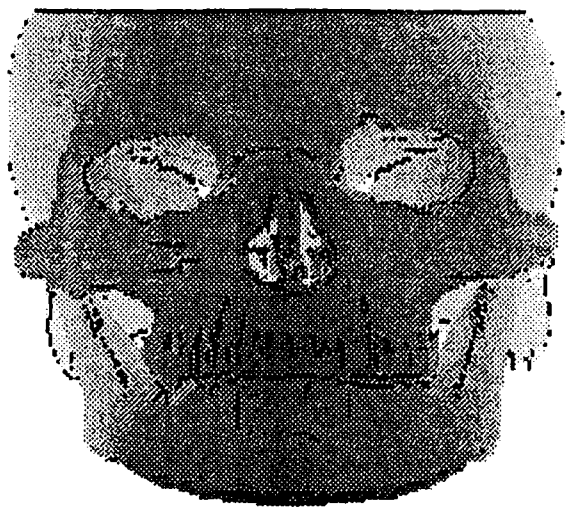


Figure 12: Face view for the position A of the local maximum curvature extrema in the maximum principal curvature direction (ridge points), The extrema are in dark and the other grey levels corresponds to the depth.



Figure 13: Face view for the position B of the local maximum curvature extrema in the principal curvature direction (ridge points), The extrema are in dark and the other grey levels corresponds to the depth.



Figure 14: Perspective view for the position A of the points where the extremality criterion is low ; The parameters of the filters are : $\alpha = 1.0$ for the first derivatives, $\alpha = 0.7$ for the second derivatives and $\alpha = 0.5$ for the third derivative



Figure 15: Results on a synthetic 2D ellipse : On left original image of the ellipse, on middle edges extracted and on left curvature extrema; The parameters of the filters are : $\alpha = 1.5$ for the smooth component, $\alpha = 1.2$ for the first derivatives, $\alpha = 1.0$ for the second derivatives and $\alpha = 0.8$ for the third derivatives. **The two false curvature extrema could be removed by a simple thresholding using the maximum curvature**

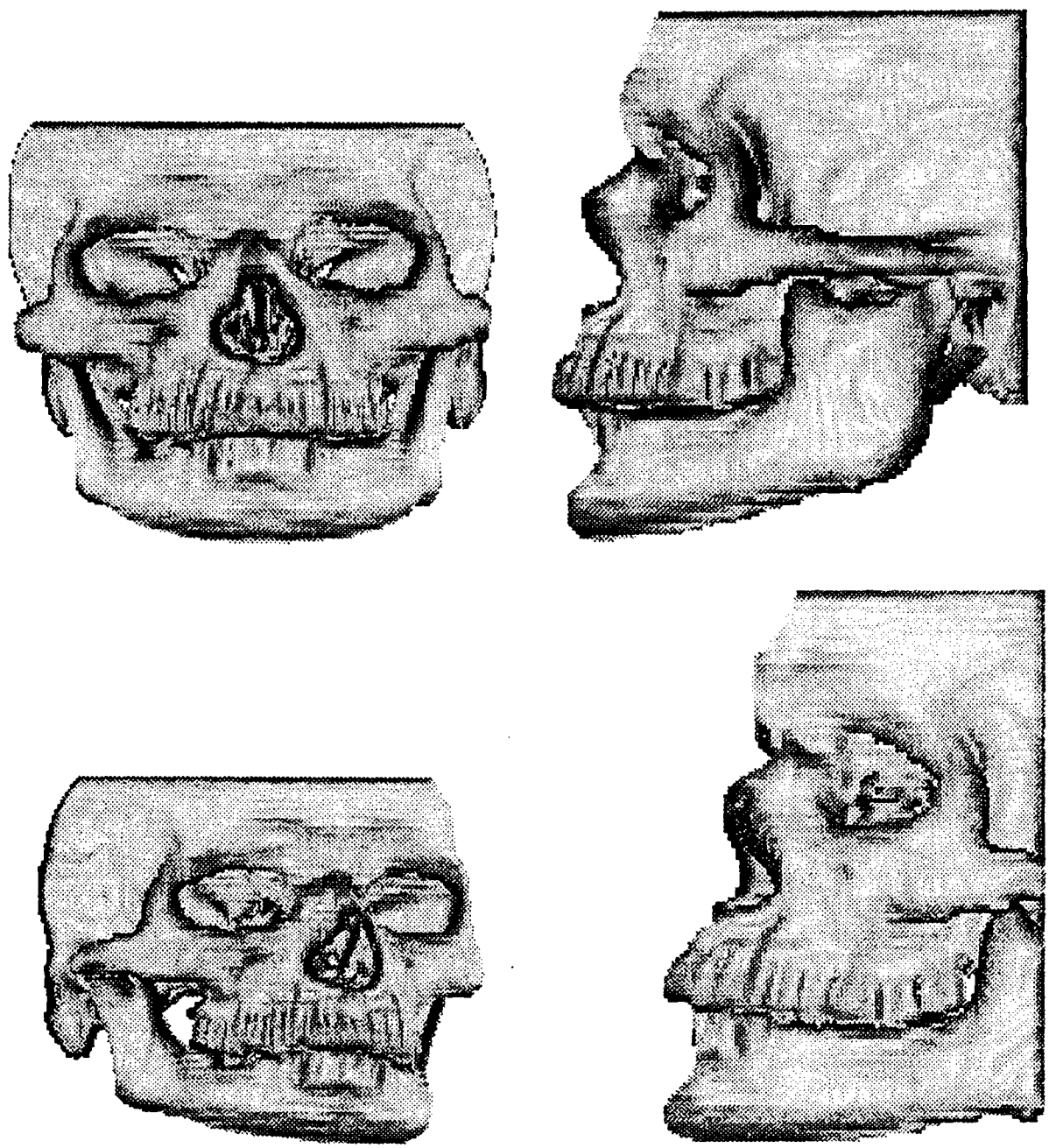


Figure 16: Perspective views of the maximum curvatures of the hypersurface for the two positions, no rendering is done : the magnitude of the maximum curvature is displayed (up left : face view for position A, up right : profile view for position A, bottom left : face view for position B, bottom right : profile view for position B). The parameters of the filters are : $\alpha = 1.0$ for the first derivatives, $\alpha = 0.7$ for the second derivatives



Figure 17: Images obtained by thresholding the maximum curvature of the hypersurface. Up : face view for the position A (the high maximum curvature points are in dark) ; bottom : face view for the position B

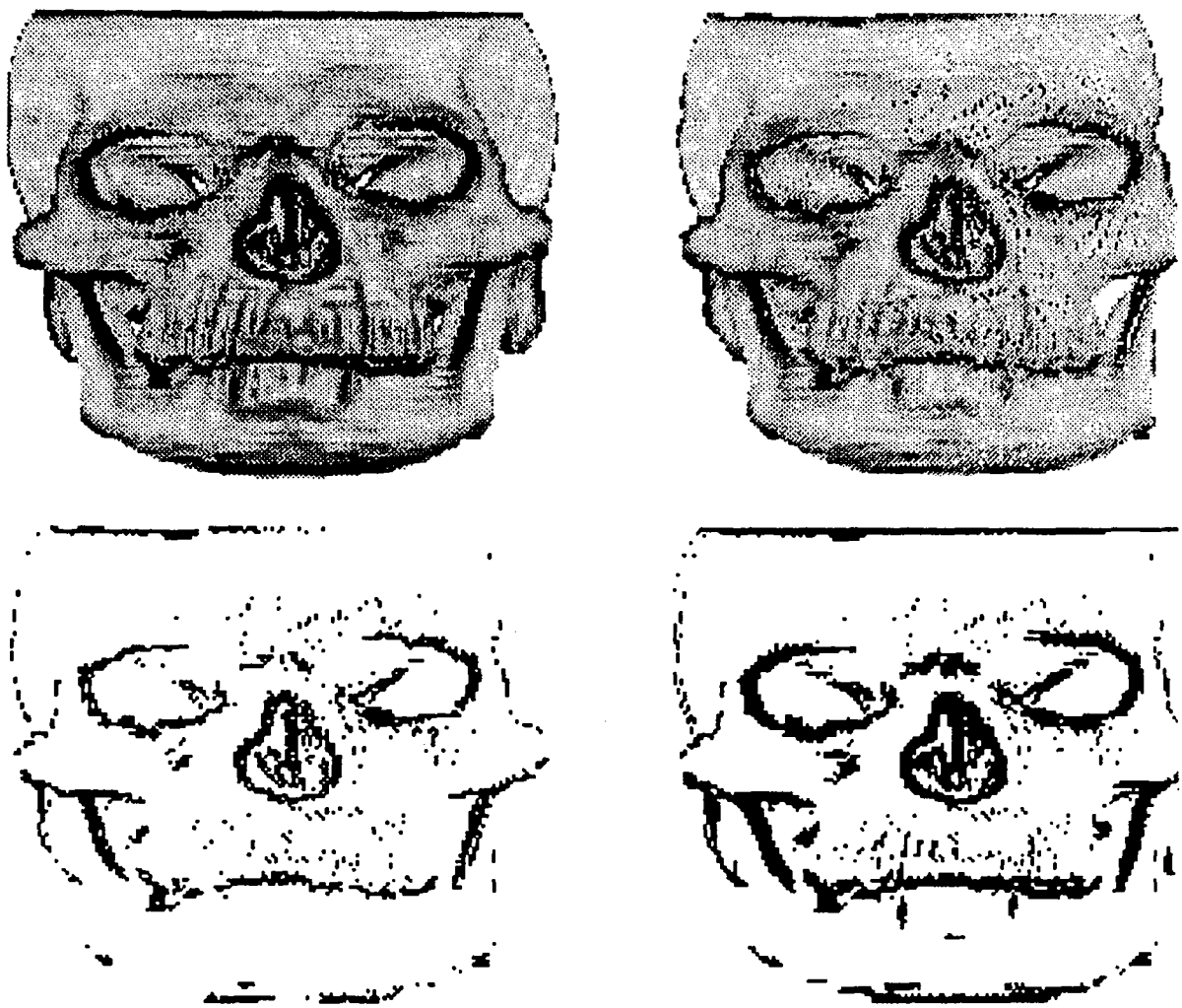


Figure 18: Up left : thresholding of the maximum curvatures for the position A, Up right : same for the position B after applying the rigid transformation (rotation and translation) from B to A, bottom right : superposition of the two results

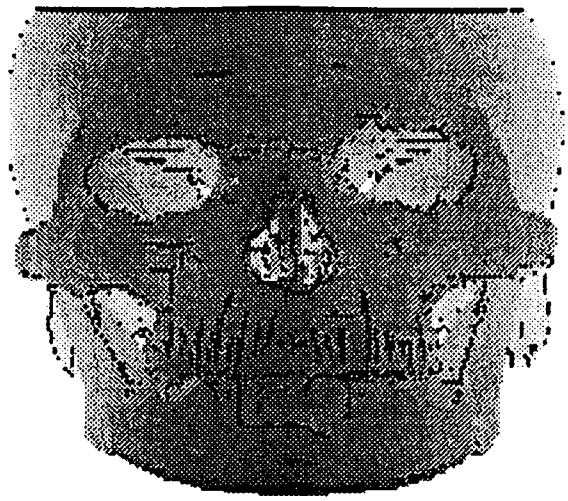


Figure 19: Face view for the position A of the local maximum curvature extrema in the principal curvature direction (ridge points), the maximum curvature of the surface is approximated by the maximum curvature of the hypersurface. The extrema are in dark and the other grey levels corresponds to the depth.

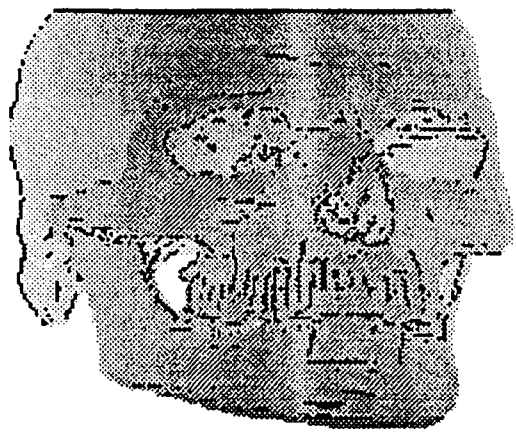


Figure 20: Face view for the position B of the local maximum curvature extrema in the principal curvature direction (ridge points), the maximum curvature of the surface is approximated by the maximum curvature of the hypersurface. The extrema are in dark and the other grey levels corresponds to the depth.

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

ISSN 0249-6399