Fast Estimation of Mean Curvature on the Surface of a 3D Discrete Object

Alexandre Lenoir

GREYC UPRESA 6072
ISMRA 6 Bd. du Maréchal Juin 14050 CAEN Cedex, France.
Email : Alexandre.Lenoir@greyc.ismra.fr

Abstract. Discrete surfaces composed of surfels (a surfel is a facet of a voxel) have interesting features. They represent the border of a discrete object and possess the Jordan property. These surfaces, although discrete by nature, represent most of the time real world continuous surfaces for which local geometrical characteristics are useful for registration, segmentation, recognition and measure. We propose a technique designed to estimate the mean curvature field of such a surface. Our approach depends on a scale parameter and has a low computational complexity. It is evaluated on synthetic surfaces, and an application is presented : the extraction of sulci on a brain surface.

Index terms - Discrete surfaces, surfels, geometric invariant, mean curvature

1 Introduction

Modern imaging techniques like MRI or confocal microscopy produce 3D digital images that represent real world scenes. A segmentation step followed by a labelling of the resulting binary image yields well identified 3D discrete objects. Their surface contains a lot of useful information. Second order differential invariants such as curvatures play a key role in image processing. For example, they are used for registration of 3D images, or for representation, recognition and segmentation of objects, as presented in [4, 7, 11]. They may also be used as internal forces that drive the deformation of an active surface. Mean curvature has a special physical signification : zero mean curvature surfaces are called minimal surfaces, because given a closed curve in 3D-space, the surface of minimal area that has this curve for border has everywhere a zero mean curvature [5] p.199. Soap films are familiar examples of minimal surfaces.

1.1 Notions of discrete surfaces and related algorithms

There are several definitions of discrete surfaces. Some are composed of subsets of voxels, seen as points of $\mathbb{Z}^3$ that verify some special topological properties [15]. Their definition is still studied in order to adapt it to intuitive notions of surfaces [3, 12]. Some other are made of surfels, that can be seen as the facets of voxels, elementary cubes whose centre has integer coordinates. This context is called the cuberille model and we will use this notion in the following. A formal description in arbitrary dimension of this model can be found in [8] and [19], and will partially be recalled in section 2. This kind of surface is still actively studied in arbitrary dimension, but their detection and properties are already well known in 3D [18,20]. Efficient techniques that give the volume enclosed by these surfaces exist [18]. Furthermore, one may also find boundary based techniques that produce the geometrical moments of an homogenous volume enclosed by the surface in [22]. Many papers have been written to propose and to justify methods that detect the connected components of surfaces in binary scenes [17,20]. However, specific techniques that estimate shape information like normal vector field, area or curvature for this surface representation seem to lack. A first step has been proposed in [10]. We use the word estimate, because from now, we will suppose that the discrete surface actually represents a smooth continuous surface, and we are interested in obtaining the geometric properties of this continuous surface. The computation of geometrical properties of the discrete surface itself is straightforward but meaningless. Moreover, as the surface of an object of practical interest is made of a big number of surfels (typically from 50 000 to 400 000), an efficient approach is welcome to avoid long computation time and to allow interactive use or multiscale analysis.

1.2 Estimation of geometric invariants

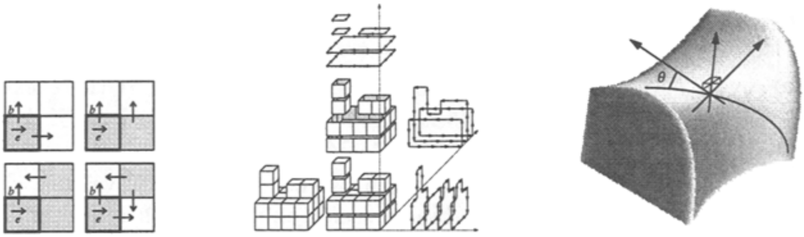
We will now briefly survey the methods that may be found in the literature and aimed at estimating the differential properties of discrete or sampled surfaces. In 2D, the estimation of differential invariants of curves like tangent, length and curvature was already well studied (see [21] and its references). The main problem of the estimation of parameters on a discrete surface (3D case) is that no parametrization of the surface exists contrarily to the 2D case. But for 2D1/2 surfaces (range images) represented by a depth map, a natural coordinate system is the 2D position. On this kind of images, one can get a local model, either by using a polynomial fit like in [4] or by computing partial derivatives of the depth function along coordinate lines. Classical formulas of differential geometry are then used to get the invariants from the partial depth derivatives or from the fitted polynomial. Different approaches exist for true 3D surface analysis. THose presented in [16] and [14] use partial derivatives of the 3D grey level image on the surface trace to compute the differential properties of the surface. These methods need to scan the entire 3D volume several times. A method presented in [13] uses the fitting of a quadratic surface patch on the surface trace. This method is to be more computationally efficient since it works only on the surface trace.

1.3 Our approach

The method described here uses the regular structure of the discrete surface as the support of functions of vectorial values that describe the geometry of the surface at the surfel scale. Then, we convolve recursively these values by a low pass filter along special curves, in order to get a regional average of these local geometrical values. These curves define coordinates curves that are common to several surfels. They are defined globally, and not locally. As the surfel belongs to two of these curves, the curves define a local parametrization of the surface. The geometric invariants are then obtained via local calculation. We use this uniform framework to compute both normals and mean curvature field. Note that the normals field is needed to get mean curvature. One of the strength of this method is its time complexity in $O(\sigma.\sqrt(n) + n)$, where σ is the scale parameter of the calculation, and n the number of surfels of the object. We can see that it is only slightly dependent on the scale parameter. This is be a attractive in a multiscale context. Another advantage is the simplicity and the uniformity of tools used to compute both normals and curvatures. The width of the neighbourhood taken into account is directly controlled by the scale parameter.

1.4 Outline of the paper

This paper is organised in the following way : in section 2 we give the basic definitions relative to discrete surfaces. Then, we explain the recursive computation of the convolution product of a summable function and a periodic function. In section 3 some basic notions of differential geometry of parametrized curves and surfaces are recalled. In section 5 we explain the calculus of mean curvature which implementation is detailed in section 4. The computational complexity is estimated in section 6 and experimental results on families of analytic objects are presented and discussed in section 7.



(a) Configurations of voxels and bels that meet the bel b at the edgel e. Grey squares are 1− voxels, white one are 0−voxels

(b) Slices and slice contours of an object along the three families of main planes. Arrows show the orientation of the contours

(c) A surface, its normal, principal curvature directions and a curve of the surface with an angle of θ with the main curvature direction.

Fig. 1. Special curves on discrete and continuous surfaces

2 Notations and basic definitions

We first describe the nature of the discrete we use, as well as the curves we consider on it. Afterwards we explain the main calculation tool used in our method: the recursive computation of a convolution product.

2.1 Surfaces of 3D discrete objects

These definitions are mainly drawn from [8] and [19] but we restrict ourselves to the 3D case.

Main vectors of $\mathbb{R}^3$, $\boldsymbol{\Delta}$, operations involving these vectors. We consider the euclidian vectorial space $\mathbb{R}^3$ and a direct orthonormal basis (O, w_1, w_2, w_3). We will use the sets of indices $I^3_+ = \{1, 2, 3\}$, $I^3 = \{-3, -2, -1, 0, 1, 2, 3\}$, and $I^3_* = I^3 \setminus \{0\}$. We define $\forall i \in I^3_+, \delta_i = \omega_i, \delta{-i} = -\delta_i$, the null vector δ_0, $\Delta = \{\delta_i / i \in I^3\}$ and $\Delta_* = \{\delta_i / i \in I^3_*\}$. We define the multiplicative operator $\otimes$ from the vector cross product $\wedge$: $\otimes : \left| {}^{I^3 \times I^3 \to I^3}_{(x,y) \to z = x \otimes y / \delta_z = \delta_x \wedge \delta_y} \right.$.

Voxel, binary scene. Let $x \in \mathbb{R}^3$. We denote x_i the ith coordinate of x. $\mathbb{R}^3$ is divided into unit cubes called *voxels* by a set of planes orthogonal to the axes $\{P_{i,j} \in I^3_+, j \in \mathbb{Z} + \frac{1}{2}$ and $P_{i,j} = \{x \in \mathbb{R}^3 / x_i = j\}\}$. We identify each voxel with its central point of $\mathbb{Z}^3$. A binary scene of $\mathbb{Z}^3$ is a function $B : \mathbb{Z}^3 \to \{0, 1\}$. We call $B(v)$ the value of the voxel v. We note $1(B) = B^{-1}(1)$ and $0(B) = B^{-1}(0)$.

$n-$adjacency, $n-$path, $n-$object, $n-$background. Two voxels c and d are said to be 6-adjacent if they share a face. They are 18$-$connected if they are 6-adjacent or if they share an edge. For $n \in \{6, 18\}$, an *$n-$path* of lenght l $p = [v_0, \dots, v_l]$ is a sequence of $l + 1$ voxels so that $\forall j \in [0, l - 1]$ v_j is $n-$adjacent to v_{j+1}. Let E be a set of voxels. Let $x, y \in E$. If there exists a $n-$path from x to y in E, we say that x and y are $n-$connected in E. A set E of voxels is $n-$connected iff any two voxels of E are $n-$connected in E. Since $n-$connectedness is an equivalence relation, we consider its equivalence classes that we call *$n-$components*. If B is a binary scene, an *$n-$object* is an $n-$connected component of $1(B)$ and an *$n-$background* is an $n-$connected component of $0(B)$.

Surfel, Surfel type, Surface, Boundary, Border, Bel. A *surfel* is an oriented surface element that is identified by the pair (v_1, v_2) of 6-adjacent voxels, of which it is the common face. Therefore,the vector $\delta = v_2 - v_1 = \delta_i \in \Delta_*$ can take six distinct values. The *type of the surfel s* is $T(s) = i$. We will call v_1 the start voxel of s, and v_2 its end voxel. A *surface* is a set of surfels. The boundary of two disjoint sets of voxels E_1 and E_2 is the set $\beta(E_1, E_2) = \{s = (a_1, a_2) / s \in \Sigma$ and $a_1 \in E_1$ and $a_2 \in E_2\}$. A *bel* of a binary scene B is a surfel $a = (c, d)$ so that $c \in 1(B)$ and $d \in 0(B)$. The boundary of a binary scene is the set of its bels. A

$\kappa\lambda$−border is the boundary of a κ−object and of a λ−background. In the sequel we will consider $\kappa\lambda$−borders, where $\kappa = 6$ and $\lambda = 18$. Such a border is proved to be connected in some sense, to have a connected interior and exterior and to possess the Jordan property.

Edgel, Edgel's type, Bel adjacency, Surface graph. We call *edgel* of the surfel a the pair $e = (a, \delta_i)$, with $\delta_i \in \Delta_*$ and $|i| \neq |T(a)|$. $T(e) = i$ is the type of e. We also say that the surfel meets the surfel at the edgel iff a and a' share an edge. We can define an adjacency relation between bels such as each edgel corresponds to a single adjacent bel. This neighbouring relation define the notion of path on a surface and of connected component of a surface, as well as the notion of surface graph. Each surfel has exactly four neighbours (one per edgel).

Slices, Slice contour, Slice contour function. A slice of $\mathbb{Z}^3$ is a set of voxels in which one coordinate is fixed, the two others being free. The slice denoted $Sl_{i,j}$ is the set of voxels whose ith coordinate is j. Let be $b = (c, d)$ a bel of type i. It belongs exactly to two *slice contours* denoted by $SlC_{i,j}, i \in I_3^+ \setminus \{|T(b)|\}$. We call i the type of the slice contour $SlC_{i,b}$. These slice contours are images of slice contour functions, denoted $SlCF_{i,b}$. The succession of slice contour bels is naturally defined by their adjacency relation and their type : $SlCF_{i,b} : \left|{}^{\mathbb{Z} \to B}_{z \to SlCF_{i,b}(z)}\right.$ with $SlCF_{i,b}(0) = b, \forall z \in \mathbb{Z}, t = T(z)$, and $e = \delta_i \wedge \omega_i$. $SlCF_{i,b}$ is then recursively defined in the following way :
$$\begin{cases} SlCF_{i,b}(z+1) \text{ is the adjacent bel of } SlCF_{i,b}(z) \text{ at } e \\ SlCF_{i,b}(z-1) \text{ is the adjacent bel of } SlCF_{i,b}(z) \text{ at } -e \end{cases}$$
Slice contours of finite objects are periodic lists of adjacent bels whose start voxels are in a same plane. A slice may contain several slice contours as shows Fig. 1(b) for horizontal slices.

2.2 Recursive computation of a discrete convolution product

Let $g : \mathbb{Z} \to \mathbb{R}$ be a convolution kernel. Let f be a periodic function of $\mathbb{Z}$ with values in $\mathbb{R}$. We want $\chi = g * f$. In [6] is explained how to approximate a gaussian kernel and its derivatives by a sum of exponential as well as the recursive implementation of a non causal filter. The k−order recursive calculation of χ is implemented by splitting g into a sum of two functions g_- et g_+, respectively null upon $\mathbb{R}^+$, and $\mathbb{R}^-_*$. For each of these functions, we need only the k previous values of the convolution products as well as the current value of f and its $k-1$ previous values. Notice that the method proposed depends on a scale parameter but that the design of the filter for any choice of scale parameter is very little time consuming. This is important since the period of the function we wish to convolve may be as small as 4 (the length of a slice contour bounding a single voxel). Whenever this case occurs it would be meaningless to keep a wide kernel for these short slice contours. In such a case, the scale of our filter is adapted so that it is always less than a fraction of the length of the slice contour it will convolve. This fraction is 4 in our implementation.

Contrary to the usual context of signal filtering, initial values of the convolution are not zero, because the function we want to convolve is not everywhere zero outside a given interval. We therefore need an initialisation step for the recursive calculation. At the start of the recursion, one can truncate g outside an appropriate interval and then compute the first k convolution products in a non recursive way. We normalise our filters such that $\sum_{x\in\mathbb{Z}} g(x) = 1$ for an averaging kernel and $\sum_{x\in\mathbb{Z}} x.g(x) = 1$ for a derivative kernel.

3 Local differential geometry

We have seen in section 2.1 that the nature of the surface defines naturally slice contours which are plane curves. If we suppose that the surface is a discretization of a smooth surface, then the slice contour is a discretized smooth plane curve. Consequently, we will recall the basic concepts of differential geometry of smooth parametrized plane curves and surfaces. These classical formulas of differential geometry of surfaces are drawn from [9].

3.1 Plane curves

Let P be a plane with a direct othonormal basis $(O; i, j)$. We set $k = i \wedge j$. Let $C : |^{U\subset R\to P}_{s\to C(s)}$ be a curve in this plane. We suppose that this curve is parametrized by its arc length:$||\frac{\delta C}{\delta s}|| = 1$. The unit tangent vector of this curve is $\tau = \frac{\delta C}{\delta s}$ and the unit normal as $\nu = \tau \wedge k$. The algebraic curvature of the curve is $\rho = \frac{\delta \tau}{\delta s}$.

3.2 Surfaces

Here we consider a smooth surface S in the euclidian 3D space (see Fig. 1(c)). The surface is parametrized in a neighbourhood of a point p by $M : |^{U\subset\mathbb{R}^2\to S}_{(u,v)\to M(u,v)}$ with $M(0,0) = p$. Let C_1 and C_2 be any two curves on the surface with $C_1(0) = C_2(0) = p$. Provided that the curves are not mutually parallel at p, the two tangents of the curves at p, $\tau_1(0)$ and $\tau_2(0)$ and are in a plane that is the tangent plane of the surface at p. N_p, the normal unit vector of T_p is colinear to $\tau_1(0) \wedge \tau_2(0)$. Let P_0 be a plane that contains N_p. Now let C_θ, $C_\theta(0) = p$ be a curve that lies both on the surface and on a plane P_θ that contains N_p. P_θ has a dihedral angle θ with P_0. Let $\rho(\theta)$ and $\tau(\theta)$ be the curvature and tangent of C_θ at p. $\rho(\theta)$ considered as of map from $[0, \pi[$ on $\mathbb{R}$ has in general two extremal values, $\lambda_1 = \rho(\theta_1)$ and $\lambda_2 = \rho(\theta_1 + \frac{\pi}{2})$. From now, we will assume that $\theta 1 = 0$. The two directions of extremal curvatures are the principal directions of the surface.

3.3 Curves on surfaces

Let C be a curve on a surface S. Let τ, ν and ρ be its tangent, normal and curvature at p. Then the curvature at p of a curve C' on S and on a plane containing N_p with the same tangent τ is $\rho_\nu = \rho.(N_p.\nu)$ and is called the normal curvature of C at p. Normal curvature depends only on the tangent of the curve and is noted $\rho(\theta)$ if $\tau = \tau(\theta)$.

4 Computing surface mean curvature from curve curvature

We explain here the principle of our method. We use slice contours and their curvatures to get the mean curvature of the surface. Slice contours are plane curves, and the computation of plane curves curvature is a common topic in the field of shape analysis. One can see [21] for a survey of curvature estimations methods. A surfel is crossed by two slice contours. We suppose we know their curvatures at each surfel. Provided the normal to the surface on the surfel s, we can have the normal curvature of each curve. We may then use the Euler formula that links the normal curvature c of a curve to the angle x of the tangent of the curve with one of the two main curvatures (λ_1 andλ_2) of the surface :

$$\text{Euler formula} \qquad c = \lambda_1 . \cos(\theta)^2 + \lambda_2 . \sin(\theta)^2 \tag{1}$$

The mean curvature is $H = \frac{\lambda_1+\lambda_2}{2}$, it is also equal to $\frac{c+c'}{2}$, when c and c' are the normal curvatures of two orthogonal curves that lie on the surface. If the curves are not orthogonal, as we do not know θ, we cannot get H. However, it is possible to get H if we know three curves $(C_i)_{i \in I_*^3}$ with normal curvatures $(c_i)_{i \in I_*^3}$ that pass through the same surface point p. First, we note that (1) can be rewritten $c = H + d.\cos(2.\theta)$ with $d = \frac{\lambda_1-\lambda_2}{2}$. For $(i,j) \in I_*^3 \times I_*^3$, we note $\alpha_{i,j}$ the angle between C_i and C_j at p. We have the set of equations $\forall i \in I_*^3, c_i = H + d.\cos(2.(\theta + \alpha_{i,j}))$ where θ is the angle between C_i and the main curvature direction. We wish to find a combination of normal curvatures in order to eliminate d for any value of θ. We use the trigonometric relation:

$$\begin{aligned} &\cos(2.\theta).\sin(2.(\alpha_{1,3} - \alpha_{1,2})) \\ &+ \cos(2.(\theta + \alpha_{1,2})) + \sin(-2.\alpha_{1,3})) \\ &+ \cos(2.(\theta + \alpha_{1,3})) + \sin(2.\alpha_{1,2}) = 0 \end{aligned} \tag{2}$$

Using the triangle relation $\alpha_{1,2} + \alpha_{1,2} + \alpha_{1,2} = \pi$, with $0 \leq \alpha_{1,2}, \alpha_{2,3}, \alpha_{3,1} \leq \frac{\pi}{2}$, we obtain $\alpha_{1,3} - \alpha_{1,2} = \alpha_{2,3} - \pi$. Then (2) can be rewritten:

$$\begin{aligned} &\cos(2.\theta).\sin(2.\alpha_{2,3}) \\ &+ \cos(2.(\theta + \alpha_{1,2})).\sin(2.\alpha_{3,1}) \\ &+ \cos(2.(\theta + \alpha_{1,3})).\sin(2.\alpha_{1,2}) = 0 \end{aligned} \tag{3}$$

The coefficients of a linear combination appear clearly and are all positive. They do not depend on θ but only on the relative angles of the curves. We define $\beta_i = \sin(2.\alpha_{j,k})$with $j \otimes = i \in I_*^3$ and $\alpha_i = \frac{\beta_i}{\sum_{i \in I_*^3} \beta_i}$. Finally we get

$$H = \sum_{i \in I_*^3} (\alpha_i . c_i) \tag{4}$$

5 Application to surfaces of discrete objects

We will now apply the previouss onto discrete surfaces. Only two curves pass trough a surfel, but if we consider a neighbourhood of a surfel we can verify that one of the following cases is true:

- One or two of the components of the normal to the surface are near zero. There exist only two types of slice contours, but they are mutually orthogonal. So we can have mean curvature from curves curvatures.
- None of the components of the normal to the surface is near zero. Slice contours are not mutually orthogonal but there exist three types of them. We may apply (4) to get the mean curvature.

We suppose from here that the normal to the surface is (a, b, c), with $a^2+b^2+c^2 = 1$. The angles between slice contours may be expressed with the coordinates of the normal to the surface. Tangent vectors of slice contours are parallel to $(-b, a, 0)$ and $(-c, 0, a)$, for slice contours of type 2 and 3. We have $\cos(\alpha_{2,3}) = \frac{bc}{\sqrt{a^2+b^2}.\sqrt{a^2+c^2}}$, and $\sin(\alpha_{2,3}) = \frac{a}{\sqrt{a^2+b^2}.\sqrt{a^2+c^2}}$. Therefore we have :

$$\beta_1 = \sin(2.\alpha_{2,3}) = 2.\cos(\alpha_{2,3}).\sin(\alpha_{2,3}) = \frac{a.b.c}{(a^2+b^2).(a^2+b^2)} \tag{5}$$

$$\text{and by definition of } \alpha_1 : \qquad \alpha_1 = \frac{b^2+c^2}{2} = \frac{1-a^2}{2} \tag{6}$$

5.1 Normal estimation

The surface graph of the 18 or 6-connected object is first computed thanks to an adaptation of surface tracking algorithm (see [1]). The whole surface graph, and not only its vertices are needed. A transversal of this graph yields the length, as well as a start surfel of each slice contour. We explained in [10] the way we compute the normals, by estimating tangents and then by doing a vector cross product, and another smoothing in order to have a good result at each surfel.

5.2 Slice contour curvature estimation

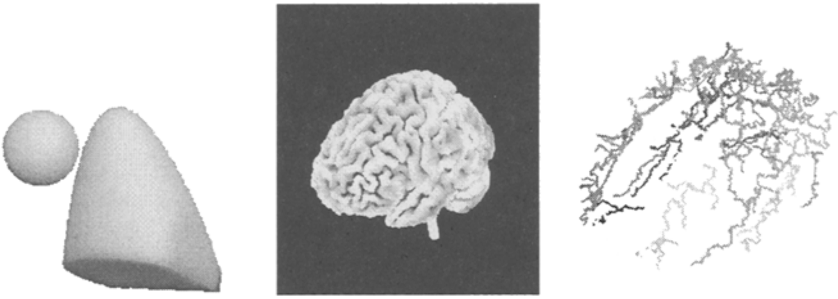
We use the tangents of each of the curves computed previously. We first normalise them and then a derivative filter along the slice contours is applied. Note that derivation is done with respect to the discrete arc length, and not with respect to the continuous arc length like in the classical definition. But as we know the ratio between the smoothed discrete arc length and the continuous arc length is the euclidian norm of the non normalised tangents, we divide the result of the derived normalised tangents by this norm. We suppose now that in a neighbourhood of a surfel s, the mean curvature varies smoothly and that the slice contour curvatures depends on the type of the slice contour but varies also smoothly inside the neighbourhood. We will note ρ_i the curvature and c_1 the normal curvature of a slice contour C_i of type i in the neighbourhood of s. For instance, for a slice of type 1, we have $c_1 = \rho_1.\sqrt{(b^2+c^2)}$.

5.3 Mean curvature estimation

Using (6) and the unit normal vector, we compute α_1 for each of the two slice contours of each surfel. We now assume that the ratio of surfels of type 2 and 3 on this slice contour are $r_{1,2}$ and $r_{1,3}$. These ratios may be estimated by a low pass filter along each curve. For a slice contour of type 1, we compute the following values : $\alpha_1.c_1 + \frac{\alpha_3.c_3}{r_{1,2}}$ or surfels of type 2 and $\alpha_1.c_1 + \frac{\alpha_2.c_2}{r_{1,3}}$ for surfels of type 3. Note that when we are on a surfel of type 2, $r_{1,2}$ cannot be zero. We recall that a surfel of type 2 intersect slice contours of types 1 and 3. Then we convolve this value with the same smoothing filter used to get the ratios and we get $\alpha_1.c_1 + \alpha_1.c_1 + \alpha_1.c_1 = H$, following (4).

6 Evaluation of algorithmic complexity

If the object is bounded by cube of size n, the number of slice contours is roughly $3.n$, and the number of surfels of its surface is around n^2 . We want to compute a field at the scale σ. The cost of the initialisation of the recursive calculation is proportional to the number of slice contours and to the scale : $O(\sigma.n)$. The recursive computation needs a constant computational amount for each surfel. There are then only local computations. As several convolutions are needed, we just have to add their costs. In conclusion, the cost of our method is $O(\sigma.n+n^2)$.



(a) A paraboloid and a sphere with identical curvatures at top.

(b) Mean curvature field on a brain surface

(c) After filtering the parts of the surface of negative mean curvature, we have extracted their centre lines

Fig. 2. Experiments and application to mean curvature field.

7 Experimental results and applications

We have made experiments in order to estimate the precision of our technique. All the experiments were done with an exponential filter with $\sigma = 4$ for averaging filters, and the derivative of an exponential filter with $\sigma = 2.5$. Filters with lower scale may have been used for short slice contours as explained in section 2.2.

7.1 Experimental results for spheres

Table 1. Experimental results for spheres

r	$\frac{1}{r}$	$H_{min}(S_r) - H_{max}(S_r)$	$\overline{\Delta}_{max}(S_r)$	$\Delta_{max}(S_r)$
5	0.2	0.248 – 0.261	30.5%	5.9%
10	0.1	0.106 – 0.107	7%	5.85%
15	0.0667	0.0678 – 0.0680	2%	8.17%
20	0.05	0.0500 – 0.0502	0.4%	10.2%
30	0.0333	0.0329 – 0.0330	1.2%	12.4%
40	0.025	0.0246 – 0.0246	1.6%	16.1%

The mean curvature of a sphere is everywhere the inverse of its radius. We have done a set of experiment on spheres of different radius, digitized one hundred times at a random position. The results of these experiments are shown in table 7.1. For a given radius r, let S_r be the set of digitized spheres of radius r. For $S \in S_r$ we denote $s(S)$ the set of its surfels and $H(s)$ the mean curvature computed at the surfel s. We define $\overline{H}(S) = \frac{1}{|S|}\sum_{s \in S} H(s)$ the mean computed value over the surfels of S. We consider $H_{\text{min}}(S_r) = \min\{\overline{H}(S), H \in S_r\}$ and $H_{\text{max}}(S_r) = \max\{\overline{H}(S), H \in S_r\}$ the extremal values for all digitized spheres of a given radius of $\overline{H}(S)$. We computed also the relative errors of mean values for all spheres $\overline{\Delta}(S_r) = \max\{|H_{\text{max}}(S_r) - \frac{1}{r}|, |H_{\in}(S_r) - \frac{1}{r}|\}.r.100$, and the maximal relative error $\Delta_{\text{max}}(S_r) = \max\{\frac{\max\{|\overline{H}(S) - H(s)|, s \in S\}}{\overline{H}(s)}\}$.

7.2 Experimental results for paraboloids

We have seen that the results for spheres were rather good. They showed that the result is quite insensitive to orientation although the data structure itself is not isotropic at all. Nevertheless, spheres do not represent all local configurations of surfaces. Indeed, every smooth surface can be approximated at the second order with its so called osculating paraboloid of equation $z^2 = \lambda_1.x^2 + \lambda_2.y^2$ in an adequate Euclidean coordinate system. Its principal curvatures are λ_1 and λ_2. This is the reason why we have done some experiments on paraboloids of various principal curvatures that represent locally any smooth surfaces. We have chosen curvatures in the range of $[-0.1, 0.1]$ with an increment of 0.025. For each paraboloid, we have discretized it one hundred times, with an arbitrary rotation. The centre of the paraboloid was chosen randomly inside a voxel. For each family of discretized paraboloid, we present in Table 7.2 four values that are the minimum, maximum, average and ideal mean curvature computed at the surfel that is the closest from the origin of the paraboloid. First it should be noticed

Table 2. Experimental results for paraboloids. In each element of this array, from top to bottom, are given the minimum value, the maximum value, the average value of the mean curvature computed at the origin of the paraboloid of equation . The last value is the ideal mean curvature

λ_2 \ λ_1	-1.00e-01	-7.50e-02	-5.00e-02	-2.50e-02	0.00e+00	2.50e-02	5.00e-02	7.50e-02	1.00e-01
-1.00e-01	-1.26e-01 -1.15e-01 -1.21e-01 -1.00e-01	-1.13e-01 -1.05e-01 -1.10e-01 -8.75e-02	-1.03e-01 -9.26e-02 -9.81e-02 -7.50e-02	-8.79e-02 -7.65e-02 -8.25e-02 -6.25e-02	-7.02e-02 -5.35e-02 -6.30e-02 -5.00e-02	-4.90e-02 -3.29e-02 -4.13e-02 -3.75e-02	-2.95e-02 -1.35e-02 -2.40e-02 -2.50e-02	-1.75e-02 -1.73e-03 -1.02e-02 -1.25e-02	-9.28e-03 1.36e-02 1.68e-03 0
-7.50e-02		-1.05e-01 -9.49e-02 -9.99e-02 -7.50e-02	-9.13e-02 -8.42e-02 -8.76e-02 -6.25e-02	-7.68e-02 -6.64e-02 -7.19e-02 -5.00e-02	-5.71e-02 -4.52e-02 -5.11e-02 -3.75e-02	-3.54e-02 -2.22e-02 -3.01e-02 -2.50e-02	-1.89e-02 -3.19e-03 -1.24e-02 -1.25e-02	-6.46e-03 7.25e-03 1.18e-03 0	2.58e-03 2.03e-02 1.25e-02 1.25e-02
-5.00e-02			-7.84e-02 -7.19e-02 -7.51e-02 -5.00e-02	-6.22e-02 -5.59e-02 -5.89e-02 -3.75e-02	-4.12e-02 -3.22e-02 -3.79e-02 -2.50e-02	-1.95e-02 -1.02e-02 -1.61e-02 -1.25e-02	-5.39e-03 7.04e-03 1.54e-03 0	7.89e-03 2.21e-02 1.57e-02 1.25e-02	1.57e-02 3.50e-02 2.70e-02 2.50e-02
-2.50e-02				-4.45e-02 -4.05e-02 -4.23e-02 -2.50e-02	-2.28e-02 -1.57e-02 -2.02e-02 -1.25e-02	-2.47e-03 5.41e-03 1.77e-03 0	1.26e-02 2.37e-02 1.97e-02 1.25e-02	2.37e-02 3.93e-02 3.38e-02 2.50e-02	3.41e-02 5.39e-02 4.51e-02 3.75e-02
0.00e+00					-8.86e-04 1.02e-02 2.76e-03 0	1.99e-02 2.79e-02 2.43e-02 1.25e-02	3.39e-02 4.51e-02 4.12e-02 2.50e-02	4.68e-02 6.05e-02 5.47e-02 3.75e-02	5.51e-02 7.24e-02 6.55e-02 5.00e-02
2.50e-02						4.13e-02 4.70e-02 4.51e-02 2.50e-02	5.86e-02 6.40e-02 6.12e-02 3.75e-02	6.89e-02 7.83e-02 7.38e-02 5.00e-02	7.86e-02 9.01e-02 8.50e-02 6.25e-02
5.00e-02							7.36e-02 7.97e-02 7.72e-02 5.00e-02	8.49e-02 9.25e-02 8.96e-02 6.25e-02	9.57e-02 1.06e-01 9.99e-02 7.50e-02
7.50e-02								9.75e-02 1.06e-01 1.02e-01 7.50e-02	1.08e-01 1.16e-01 1.12e-01 8.75e-02
1.00e-01									1.18e-01 1.27e-01 1.22e-01 1.00e-01

that the results are more biased than for the sphere. They are overestimated in term of absolute value, but corresponds well to intuitive estimation of curvature that is less "local" than its mathematical definition (Figure 2(a) illustrates his point). Nevertheless, they still be quite insensitive to translation and rotation. The sign of the mean curvature is also well preserved.

8 Conclusion and perspectives

We have described in this paper an original and efficient technique that estimate the mean curvature field of a discrete surface. It depends on a scale parameter. Both time and space complexity is linear with respect to the number of surfels. Our method is fast enough to be useful in an interactive tool of manipulation of discrete surfaces, or in a multi-scale context. Our technique can be applied to object recognition and surface segmentation purposes. It may even be appropriate for measure when the surfaces curvature is regionally constant. Since the result strongly depends on the estimation of the slice contour curvature, further experiments with other methods of discrete plane contour curvature estimation should be done. The method of computation of differential characteristics may be adapted to work in higher dimensional spaces, to characterise the shape of discrete hypersurfaces since surface definition as well as slice contours still exist in these spaces. As an application, we have considered the surface of the human brain as a classical 2D grey level image where the grey level is the mean curvature and have extracted characteristic surface parts (Fig. 2(b) and Fig. 2(c)).

References

1. E. Artzy, G. Frieder, and G. T. Herman. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *CVGIP*, 15(6):1–24, 1981.
2. G. Bertrand and R. Malgouyres. Some topological properties of discrete surfaces. In *Proceedings of DGCI'96*, volume 1176 of *Lecture Notes in Computer Sciences*. Springer, 1996.
3. P. J. Besl and R.C. Jain. Invariant surface characteristic for 3d object recognition in range images. *CVGIP*, 33(999):33–80, 1986.
4. M. Do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.
5. R. Deriche. Recursively implementing the gaussian and its derivatives. Rapport de Recherche 1893, INRIA, April 1993.
6. T. J. Fan, G. Medioni, and R. Nevata. Recognising 3d objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1140–1157, November 1989.
7. G.T. Herman. Discrete multidimensional jordan surfaces. *CVGIP : Graphical Models and Image Processing*, 54(6):507–515, November 1992.
8. Y. Kerbrat and J.M. Braemer. *Géométrie des courbes et des surfaces*. Collection méthodes. Hermann, 1976.
9. A. Lenoir, R. Malgouyres, and M. Revenu. Fast computation of the normal vector field of the surface of a 3d discrete object. In *Proceedings of DGCI'96*, volume 1176 of *Lecture Notes in Computer Sciences*, pages 101–112. Springer, 1996.
10. P. Liang and C. H. Taubes. Orientation based differential geometric representations for computer vision applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3), March 1994.
11. R. Malgouyres. A definition of surfaces in $\mathbb{Z}^3$. *Theorical Computer Sciences*. to appear.
12. O. Monga, N. Ayache, and P. T Sander. From voxel to curvature. Rapport de Recherche 1356, INRIA, December 1990.
13. O. Monga and S. Benayoun. Using partial derivatives of 3d images to extract typical surface features. *CVIU*, 61:171–189, 1995.
14. D.G. Morgenthaler and A. Rosenfeld. Surfaces in three-dimensional digital images. *Information and Control*, 51(3):227–247, 1981.
15. J. P. Thirion and A. Gourdon. Computing the differential characteristics of isointensity surfaces. *CVIU*, 61:190–202, 1995.
16. L. Thurfjellj, E. Bengtsson, and Bo Nordin. A boundary approach for fast neighbourhood operations on three- dimensional binary data. *CVGIP*, 57(1):13–19, 1995.
17. J. K. Udupa. Determination of 3d shape parameters from boundary information. *Computer Graphics and Image Processing*, 17:52–59, 1981.
18. J. K. Udupa. Multidimensional digital boundaries. *CVGIP: Graphical Models and Image Processing*, 56(4):311–323, July 1994.
19. J. K. Udupa and V .G. Ajjanagadde. Boundary and objet labelling in three-dimensional images. *CVGIP*, 51:355–369, 1990.
20. M. Worring and A.W.M. Smeulders. Digital curvature estimation. *CVGIP IU*, 58(3):366–382, 1993.
21. L. Yang, F. Albregtsen, and T. Taxt. Fast computation of three-dimensional geometric moments using a discrete divergence theorem and a generalisation to higher dimensions. *CVGIP*, 59(2):97–108, March 1997.