MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

A MULTI-SCALE APPROACH FOR CREST LINE EXTRACTION IN 3D MEDICAL IMAGES

Olivier Monga, Richard Lengagne, Rachid Deriche*, Song De Ma†

INRIA, Domaine de Voluceau - Rocquencourt
BP. 105, 78153 Le Chesnay Cedex, France
NLPR, Institute of Automation, Academia Sinica
P.O. Box 2728, Beijing 100080, China
email: lengagne@prlsun3.ia.ac.cn

ABSTRACT

Recently, we have shown that the differential properties of the surfaces represented by 3D volumic images can be recovered using their partial derivatives. For instance, the crest lines can be characterized by the first, second and third partial derivatives of the grey level function $I(x, y, z)$. This paper deals with the following points :

- the computation of the partial derivatives of an image can be improved using recursive filters which approximate the Gaussian filter,
- a multi-scale approach solves many of the instability problems arising from the computation of the partial derivatives,
- we illustrate the previous point for the crest line extraction (a crest point is a zero-crossing of the derivative of the maximum curvature along the maximum curvature direction).

We present experimental results of crest point extraction on 3-D medical data.

keywords : *volumic 3D medical images, surface modelling, curvatures, crest lines, multi-scale derivation, recursive filtering.*

I INTRODUCTION

Volumic 3D images are now widely distributed in the medical field. They are produced from various modalities such as Magnetic Resonance Imagery (MRI), Computed Tomography Imagery (CT), Nuclear Medicine Imagery (NMI) or Ultrasound Imagery (USI). Such data are represented by a discrete 3D grey level function $I(i, j, k)$ where the high-contrast points (3D edge points) correspond to the discrete trace of the surfaces of the geometrical structures [3]. A motivating issue is then to extract typical features of these surfaces. The most natural way is to look for differential Euclidean surface invariants such as : curvatures, crest lines, parabolic lines, umbilic points... [2, 5, 6]. Recently, we have shown that the differential properties of a surface defined by an iso-contour in a 3D image can be recovered from the partial derivatives of the corresponding grey level function [2]. In [2] crest lines are extracted using first, second and third order partial derivatives provided by 3D Deriche filters [3]. The critical point of this approach also studied in [6] is the stability of expressions including second and third order partial derivatives such as the "extremality criterion" defined in [2, 6].

In this paper we propose isotropic recursive 3D filters to improve the computation of partial derivatives and also a multi-scale approach to extract the zero-crossings of the extremality criterion in a better way.

Section II recalls the main results of [4] about the interest of using isotropic filters to compute differential Euclidean invariants.

Section III deals with the computation of the curvatures of the surfaces traced by the iso-contours (3D edge points) from the partial derivatives of the image (for instance provided by the previous method), using the main results of the reference [2] and shows the problems induced by a single scale filtering.

In Section IV, we propose to use different widths of filters to compute the curvatures. This leads to a multi-scale curvature computation scheme. We apply this principle to track the zero-crossings of the derivative of the maximum curvature points along the maximum curvature direction (extremality criterion) which correspond to the crest points. The zero-crossings coming from the different scales are merged using a valuated adjacency graph. Simple and efficient strategies to extract stable zero-crossings from this graph, are proposed. Another method using adaptive filtering is also proposed.

In Section V we present experimental results obtained on real data (a CT 3D image). We show that our approach combining a multi-scale scheme and also the use of isotropic filters provides reliable crest lines even for noisy data.

II COMPUTATION OF THE PARTIAL DERIVATIVES OF A 3D IMAGE USING LINEAR FILTERS

II.1 Recursive and isotropic filtering

We show in [4] that isotropic filters are required to calculate properly differential Euclidean invariants from images: if the partial derivatives of an images are computed using filters derived from an isotropic smoothing filter, then the differential Euclidean invariants calculated using these partial derivatives are also invariants by a rigid motion (Euclidean invariants).

The reference [2] uses Deriche's smoothing operator : $f(x, y, z) = f_0(x)f_0(y)f_0(z)$ with $f_0(x) = c_0(1 + \alpha \mid x \mid)e^{-\alpha|x|}$ where c_0 is a normalization constant.

The advantage of using this function is that we obtain recursive filters which are optimal for Canny's criteria in the direction of the frame axis X, Y, Z. The drawback is that our 3D smoothing operator $f(x, y, z)$ is not isotropic i.e. not rotationally invariant. For instance, this implies that the magnitude of the gradient, or the curvatures, computed with the corresponding derivative filters are not invariant by a rigid motion. On the other hand, we also take interest in using separable recursive filters in order to obtain a reasonable computational cost. A way to join these two antagonist points is to use the recursive approximation of the Gaussian filter (the only separable non trivial smoothing filter) introduced by R. Deriche in the recent reference [1].

The 1D Gaussian smoothing filter is : $G(x) = e^{-\frac{x^2}{2\sigma^2}}$
Using Prony's method described in [1], the positive and negative part of G and of its normalized derivatives of first and second order can be approximated by a 4th order recursive operator (IIR):

*At INRIA Sophia-Antipolis, 06565 Valbonne Cedex, France.

†This work was partially supported by the Sino-French cooperation program on medical imaging.

$$h(x) = (a_0 \cos(\frac{\omega_0}{\sigma}x) + a_1 \sin(\frac{\omega_0}{\sigma}x))e^{-\frac{b_0}{\sigma}x} +$$
$$(c_0 \cos(\frac{\omega_1}{\sigma}x) + c_1 \sin(\frac{\omega_1}{\sigma}x))e^{-\frac{b_1}{\sigma}x}$$

We also extend this filtering scheme to the third order derivative and to the 3D case. We develop a set of recursive filters approximating the Gaussian and its derivatives which can be used to compute the first, second, and third order derivatives of a 3D image.

We stress that a very important point not carried out in [1] is the normalization of the filters which allows to obtain coherent values for the different derivatives. Here we use the scheme presented in [2] to compute the normalization constants. All the details about the normalization constants and the recursive implementation of the filters can be found in [4].

II.2 Algorithm to compute the first, second and third derivatives of a 3D volumic image

We obtain for the computation of $\frac{\partial^n g}{\partial x^m \partial y^p \partial z^q}$, $m+p+q=3$ the following algorithm where the convolution products are implemented using the recursive implementation of the filters :

$$\text{for } (m,p,q)/(m+p+q) \leq 3 \text{ do}$$
$$\left[\begin{array}{l} R = I \\ R = R * g_m(x) \\ R = R * g_p(y) \\ I_{x^m y^p z^q} = R * g_q(z) \end{array}\right.$$

III USING THE PARTIAL DERIVATIVES TO EXTRACT AND TO CHARACTERIZE 3D STRUCTURES OF A VOLUMIC IMAGE

III.1 3D edges, curvatures and ridge points from partial derivatives

Classically, 3D edge detection can be done by computing the first derivatives (gradient approach) or the Laplacian (Laplacian approach) of the image. Recently, recursive filtering has been introduced to define 3D edge detection operators having a better noise immunity and a lower computational cost [3]. Instead of the recursive filters proposed in [3], we use the above-mentioned filters to ensure the invariance of the features under rigid motion. Then we use the main results of the reference [2], to compute the curvatures, the principal curvature directions and the curvature derivatives with differential geometry formulas.

III.2 Practical computation of the crest lines of the surfaces in a 3D image at a given resolution

The main stages of our algorithm allowing to extract crest lines in a 3D image are :

1. Computation of the first, second and third order partial derivatives of the image $I(x,y,z)$ ($\frac{\partial^n f}{\partial x^m \partial y^p \partial z^q}$, $m+p+q=3$) using the recursive filters defined in Section II for a given value of σ ;
2. Extraction of the 3D edge points using the first order partial derivatives (gradient) of I ;
3. For each point of the 3D edge map, computation of :
   - the two principal curvatures and the corresponding principal curvature directions;
   - the extremality criterion (derivative of the maximum curvature along the corresponding principal direction).
4. Building of an extremality criterion image $C_\sigma(x,y,z)$ such as at each edge point (x,y,z), $C_\sigma(x,y,z)$ is set to the value of the extremality criterion and to 0 otherwise;
5. Determination of an image $Z_\sigma(x,y,z)$ set to 1 at each edge point being a zero-crossing of the extremality criterion and to 0 otherwise.

The last stage of this algorithm consists of finding the zero-crossings of a function defined on the discrete trace of a surface (traced by the 3D edge points) which is a difficult task in itself. So far, we have only implemented simple strategies to extract these zero-crossings. But, in order to be solved properly, this delicate problem needs more attention. An interesting solution can be found in [6].

Therefore, the final output of our algorithm is an image Z_σ representing the set of edge points which are zero-crossings of the extremality criterion. Each value of σ defines an image Z_σ representing the crest line for the scale defined by σ.

IV MULTI-SCALE APPROACH TO EXTRACT CREST LINES IN 3D VOLUMIC IMAGES

IV.1 Why a multi-scale approach ?

As we have seen in the previous section the result of our algorithms is an image Z_σ where the zero-crossings of the extremality criterion are marked. Thus, Z_σ shows the crest points for the scale defined by σ. Generally, we see that :

- for simple data, we can obtain good results using a single value for σ but the problem of the correct choice of this value is still open.
- for more complex data the suitable value for σ varies depending on the area of the 3D image ;
- for noisy data, only the crest lines that can be seen using different scales define reliable features.

Therefore, similar to the edge detection [7] and to the crest line extraction in depth maps [5], it is of great interest to use a multi-scale approach. Moreover the recursive implementation of our filters makes it reasonable in terms of computational cost.

IV.2 Merging the results using a Multi-scale Adjacency Graph

In order to merge the results obtained at different scales $\sigma_i, i = 1, n$ we propose a practical and efficient data structure that we will call the Multi-scale Adjacency Graph : $G_{\sigma_1,\sigma_2,\ldots\sigma_n}$. $G_{\sigma_1,\sigma_2,\ldots\sigma_n}$ is a valuated graph built as follows :

1. each node of $G_{\sigma_1,\sigma_2,\ldots\sigma_n}$ is attached to a point (i,j,k) such that for at least one scale σ_m we have $Z_{\sigma_m}(i,j,k) = 1$;
2. the features attached to each node are :
   (a) the coordinates of the corresponding 3D point $((i,j,k))$;
   (b) the values of the scales for which this point is a crest point (all the σ_p such that $Z_{\sigma_p}(i,j,k) = 1$);
   (c) the differential characteristics extracted for all the scales : principal curvatures and principal curvatures directions, value of the extremality criterion.
3. we define an edge joining two nodes of $G_{\sigma_1,\sigma_2,\ldots\sigma_n}$ if and only if the two corresponding points are adjacent for the 26-connectivity ;

Therefore $G_{\sigma_1,\sigma_2,...\sigma_n}$ represents the results of the crest point extraction for the different scales and their spatial relationships. This data structure is particularly efficient when the stability of the crest point locations through different scales is a good selection criterion. Our experiments performed on real and synthetic data show that generally the position of the reliable crest points remain the same for different values of the scale σ (i.e. the shifts of the crest points are less than one pixel).

For instance, the following simple pruning strategy for the graph $G_{\sigma_1,\sigma_2,...\sigma_n}$ can be used :

1. select all nodes corresponding to points which are crest points for at least a given number of scales ;
2. select the connected components having at least a given number of nodes (this threshold corresponds to the minimal number of points of a crest line).

We come up with the following algorithm :

1. Computation of the zero-crossings of the extremality criterion for a given set of scales : $\sigma_1, \sigma_2, ...\sigma_n$; the result is a set of images $Z_{\sigma_1}, Z_{\sigma_2}...Z_{\sigma_n}$;
2. building of the multi-scale graph $G_{\sigma_1,\sigma_2,...\sigma_n}$ (see section IV.2) ;
3. pruning of $G_{\sigma_1,\sigma_2,...\sigma_n}$ to select reliable crest points.

IV.3 A multi-scale approach using adaptive filtering

The drawback of the previous method is that, in order to compute the partial derivatives, the same scale is used at each point, regardless to the curvature value at this point. Intuitively, we should use large scales when the curvature is small and smaller scales when the curvature is large. Moreover, some simple computations show that the computation of the curvature (with the method described in Section II) at some points of a circle of a given radius R can lead to errors from 20 to 2000 % if the filter-width σ is $R/50$, from 0.04 to 40 % if $\sigma = R/10$, from 0.04 to 7.2% if $\sigma = R/4$, according to the point where the curvature is computed. It thus seems that the error is all the smaller as the filter-width is larger. However, this width cannot be too large so that the smoothing of one part of the shape (say, here, a circle) does not influence the other parts.

A first estimation of the curvature (or the principal curvatures in the 3D case) can be done with a "reasonable" scale: in the following experiments, this scale is set to $T/8$, where T is the smallest of the image dimensions. Since the shape is roughly a sphere and it is approximately symmetric, this corresponds to $\sigma = R/4$, where R is the radius of the sphere. In that case, the error in the computation of the curvatures is rather small. Then, the computation of the partial derivatives and the extraction of the zero-crossings of the curvature derivative is launched for a given number of scales; the range of the scales is defined by the minimum and the maximum of the curvature values throughout the image. According to the previously estimated curvature value K_0 at a given point, we select the "appropriate scale" at this point and build the image of the zero-crossings. Practically, the "appropriate scale" is the closest one to $1/(4 * K_0)$, for the same reasons as explained before. Therefore, this method is a kind of adaptive filtering where the value of the filter-width depends on the shape of the object.

V EXPERIMENTAL RESULTS

We present experimental results obtained on real data from the implementation of the algorithms described in the previous section.

We have tested our method on two 3D X-ray scanner images of the same skull taken at two different positions (see figures 1 to 6). We have extracted the maxima of the maximum curvature in the maximum curvature direction. The stability of the results we obtain for a single scale illustrates the rotation invariance of our computation of the curvatures and of the extremality criterion (see section II). We first show the crest point extraction using the mono-scale approach ($\sigma = 5$, for a 64*64*64 image), then the thresholding of the multi-scale image (only the points appearing at at least 3 scales out of 10 are displayed) and finally the first results obtained with the adaptive filtering. All these results can then be improved using the graph structure as above-mentioned. More experimental results on synthetic and real data can be found in the reference [4]. We point out that the size of the convolution mask for a direct implementation of a 3D Gaussian of variance σ^2 is $(8\sigma)^3$ (for $\sigma = 4$ we obtain 8192 !). The use of recursive filters of order 4 reduces this computational cost to about 100 operations per point for any value of σ. Of course, the previous remark applies also for the derivatives of the gaussian filter. Therefore, even for a single space scheme, the recursive filtering appears as a crucial tool.

VI CONCLUSION

We have presented a multi-scale approach to extract crest lines of the surfaces represented by 3D volumic images. Compared to the method described in [2] we have developed the following points :

- we show the great theoretical interest in using filters derived from an isotropic smoothing filter to compute partial derivatives of an image ;
- we propose to use recursive filters approximating the Gaussian and its derivatives to obtain differential characteristics invariant by rigid motion ;
- in order to improve the stability of the computation of the differential characteristics (curvatures, derivative of the curvature) we use a multi-scale approach.

We stress that the same sketch could be used to extract other differential singularities such as : parabolic lines, umbilic points... Besides, this methodology could also be used in 2-D images like interior scenes, to extract corners for instance.

References

[1] Rachid Deriche. Recursively implementing the gaussian and its derivatives. *INRIA Report research*, 1993.

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

[3] 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.

[4] Olivier Monga, Richard Lengagne, and Rachid Deriche. Extraction of the zero-crossings of the curvature derivative in volumic 3D medical images: a multi-scale approach. *INRIA Research report*, 1994.

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

[6] J-P. Thirion and Gourdon A. The 3d marching lines algorithm and its application to crest lines extraction. Technical Report 1672, INRIA, May 1992.

[7] A. Witkin. A multi-scale approach to extract zero-crossings of the laplacian. In *Proceedings of International Joint Conference on Artificial Intelligence*, 1982.

Figure 1: mono-scale approach: the crest points present at the scale σ=5 are marked.

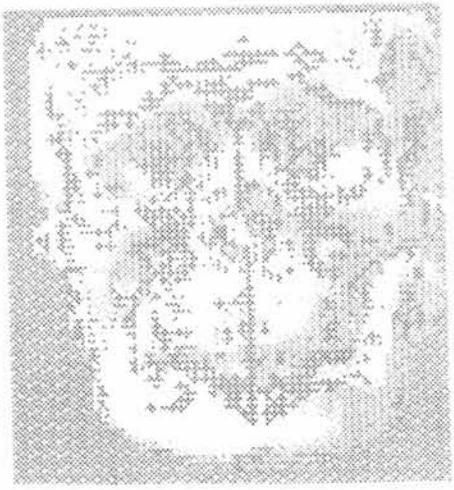

Figure 2: multi-scale approach: the crest points present at at least 3 scales out of 10 ($\sigma = 1..10$) are marked.

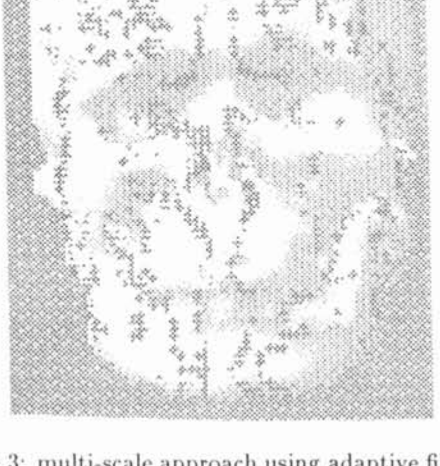

Figure 3: multi-scale approach using adaptive filtering: the scale at each point varies accordingly to the first estimation of the curvatures.

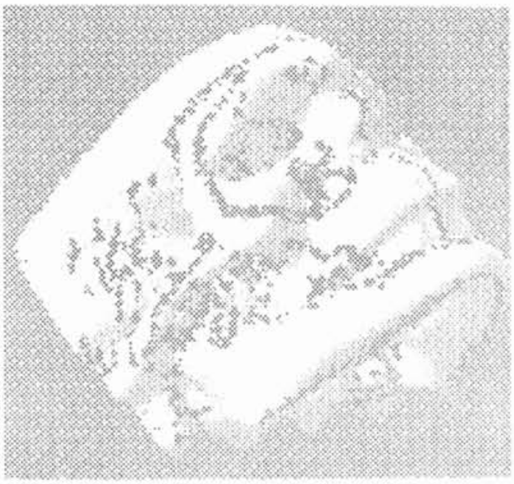

Figure 4: same as fig. 1 in another position.

Figure 5: same as fig. 2 in another position.

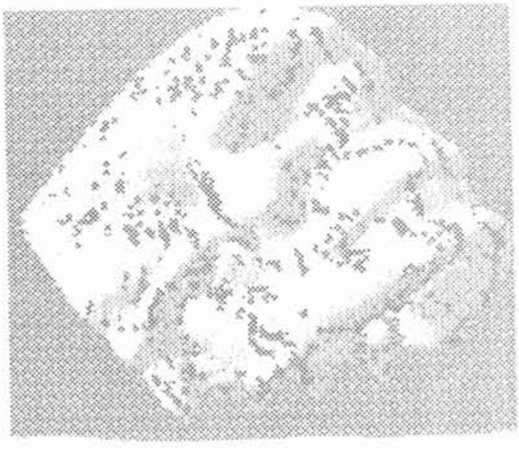

Figure 6: same as fig. 3 in another position.