Recursive filtering and edge closing: two primary tools for 3d edge detection

Olivier Monga, Rachid Deriche, Grégoire Malandain, J.P. Cocquerez

INRIA

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

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

Rapports de Recherche

N° 1103

Programme 6
Robotique, Image et Vision

RECURSIVE FILTERING AND EDGE CLOSING : TWO PRIMARY TOOLS FOR 3d EDGE DETECTION

Olivier MONGA
Rachid DERICHE
Grégroire MALANDAIN
Jean-Pierre COCQUEREZ

Octobre 1989

RR-1103

Recursive Filtering and Edge Closing : two primary tools for 3d edge detection

Olivier MONGA
INRIA - Domaine de Voluceau - B.P. 105
78153 LE CHESNAY CEDEX - FRANCE

Rachid DERICHE
INRIA Sophia-Antipolis - 2004, Route des Lucioles
06565 VALBONNE CEDEX - FRANCE

Grégoire MALANDAIN
INRIA - Domaine de Voluceau - B.P. 105
78153 LE CHESNAY CEDEX - FRANCE

Jean Pierre COCQUEREZ
ENSEA - Impasse des chênes pourpres
95014 CERGY PONTOISE - FRANCE

Abstract

This paper deals with edge detection in 3d images such as scanner, magnetic resonance (NMR), or spatio-temporal data. We propose an unified formalism for 3d edge detection using optimal, recursive and separable filters recently introduced for 2d edge detection. Then we obtain some efficient 3d edge detection algorithms having a low computational cost. We also show that 3d edge closing enables to extract many edges not provided by the filtering stage without introducing noisy edges. Experimental results obtained on NMR images are shown.

Filtrage récursif et fermeture de contours : deux outils essentiels pour la détection de contours 3D

Résumé

Ce rapport est consacré à la détection de contours dans des images tridimensionnelles (scanner X, résonnance magnétique nucléaire, images spatiotemporelles ...). Il fait suite au rapport de recherche INRIA No. 930 de Novembre 1988 qui présente un exemple de filtre récursif pour l'approximation du gradient 3D. Nous proposons ici un formalisme unifié pour la détection de contours 3D en utilisant des filtres récursifs et séparables pour approximer le gradient ou le laplacien.

Nous montrons aussi que la fermeture de contours 3D permet d'extraire des contours non fournis par la partie filtrage sans introduire de contours bruités.

On présente des résultats expérimentaux sur des images médicales 3D obtenues par résonnance magnétique nucléaire (RMN).

1 Introduction

Many industrial or medical applications provide three dimensional images representing volumic informations. Modern scanning techniques such as Computed Tomography (CT) produce 3d images where the grey level function is proportional to the local density [5]. In biomedicine for example, 3d images are produced by magnetic resonance imaging (NMR), computed tomography (CT), or positron emission tomography (PET). Thus, in such data the volumes presenting a homogeneous grey level distribution correspond to the various entities of the 3d structure. To extract these volumes we can either look directly for a partition of the 3d image into homogeneous area : region based approach either search the surfaces forming their boundaries : edge based approach. The segmentation into regions of 2d or 3d images set the acute problem of finding some homogeneity properties suitable for the regions [7]. This is the main reason why we have chosen an edge based approach that can be split into two main stages : 3d edge detection by filtering and 3d edge closing using morphological properties.

Then a first stage to identify these surfaces is to search their points using only signal information. The points of such surfaces are located where the gradient of the grey level function is locally maximum. This set the classical edge detection problem widely studied for 2d images [1,3,2], but which has not yet received much attention in 3d. Basically 2d and 3d edge detection set the same problem ie how to detect the discontinuities of a discrete and noisy (2d or 3d) function.

Thus existing 3d edge detectors are issued from a generalization in 3d of 2d edge detectors [11,6,13,10,9]. Their aim is to approximate the gradient or the laplacian of the image thanks to convolution masks. Then there is a trade-off to meet between the size of the convolution masks and their detection and localisation performances. Particularly in the 3d case the huge size of data makes the algorithmic complexity and the storage requirement be key points for 3d edge detection. Thus the size of the convolution masks used to implement the operator cannot be too large in order to avoid very prohibitive computing times. The convolution of a 3d image $dimx \times dimy \times dimz$ by a mask $n \times n \times n$ costs $(n^3 dimxdimydimz)$. For instance, this complexity makes almost impossible to use sizes of masks greater than $5 \times 5 \times 5$ for an image $256 \times 256 \times 64$. Therefore an important limitation of many 3d edge detectors has been the use of small convolution masks very sensitive to noise.

Recently in order to get rid of that drawback recursive filtering has been introduced for 3d edge detection [8]. It allows to implement filters having an infinite impulse response with a low computational cost (about the same than a 3.3.3 convolution mask in the 3d case). The 3d edge detection operator proposed in [8] is issued from a 2d operator that is extended to 3d thanks to separable filtering [3].

In this paper we propose an unified formalism for 3d edge detection using recursive filters either for the extraction of gradient extrema, either for the determination of the laplacian zero crossing. This is achieved thanks to a direct extension of the 2d algorithms [2] for the first derivative approach. In the case of the second derivative approach the extension is less direct.

Due to the noise this filtering stage is generally not sufficient to obtain results to be used for surface and volume reconstruction. To improve the 3d edge map it is therefore useful to introduce morphological informations. We present a 3d edges closing algorithm using the following assumptions :

- on each planar cross section parallel to XY, XZ, and YZ the edges define closed contours
- it dœs not exist holes of width 1 between two sections parallel to XY, XZ and YZ

We show that this edge closing method enables to improve significatly the results obtained by the filtering stage.

The article is organized as follow :

In section 2 we show that separable filtering enables to reduce the nd (and especially 3d) edge detection problem to smoothing and derivation of a 1d signal. This is true both for first and second derivative approaches.

In section 3 we list some 1d optimal [2] and recursive filters aiming to smooth or to derive a 1d signal. Section 4 describes the algorithm in the 3d case. Section 5 deals with the 3d edges tracking/closing method. In sections 5 and 6 we present experimental results obtained on NMR data and conclude.

2 Statement of the nd edge detection problem

The goal of this section is to show that under reasonable assumptions, the nd edge detection (and particularly 3d) task can be reduced to smoothing and derivation of a 1d signal.

Let $I(x_1, x_2, \ldots, x_n)$ be a noisy signal of dimension n

Classicaly edge detection is tackled thanks to the following two approaches :

1. Gradient computing and extraction of the local extrema of the gradient magnitude in the gradient direction : Gradient approach.
2. Laplacian computing and determination of the zero crossing : Laplacian approach

Let $G(I)$ be the gradient of I :

$$G(I) = (\frac{\partial I}{\partial x_1}, \frac{\partial I}{\partial x_2}, \ldots, \frac{\partial I}{\partial x_n})^t$$

Let $L(I)$ be the laplacian of I :

$$L(I) = \frac{\partial^2 I}{\partial {x_1}^2} + \frac{\partial^2 I}{\partial {x_2}^2} + \cdots + \frac{\partial^2 I}{\partial {x_n}^2}$$

A classical approach is to search linear filters to approximate $G(I)$ and $L(I)$. Given that the derivative of a convolution product is equal to the convolution of the signal by the derivative of the filter response, the approaches 1 and 2 can be reduced to the determination

of a smoothing filter $L(x_1, x_2, \ldots, x_n)$. First and second derivatives with respect to x_i are respectively computed by convolving the image with $\frac{\partial L(x_1,x_2,\ldots,x_n)}{\partial x_i}$ and $\frac{\partial^2 L(x_1,x_2,\ldots,x_n)}{\partial {x_i}^2}$. We notice that the laplacian can be directly computed by convolution with the filter whose response is : $\frac{\partial^2 L(x_1,\ldots,x_n)}{\partial {x_1}^2} + \cdots + \frac{\partial^2 L(x_1,\ldots,x_n)}{\partial {x_n}^2}$. We will see in the next section that in some cases this expression can be simplified. Another solution for the laplacian computing consists in approximating it by difference between the image smoothed by two filters of different parameters [12]. However the results obtained by this solution are less satisfactory than these provided by a direct laplacian computing.

The definition of convolution masks to approximate gradient or laplacian set an acute computing time problem. For example in dimension 2 the convolution of an image $d_x \times d_y$ by a convolution mask of size $p \times p$ costs $p^2 d_x d_y$. This leads to the use of filers having separable response with respect to directions $x_1, x_2, \ldots, x_n$:

$$L(x_1, x_2, \ldots, x_n) = L_{x_1}(x_1) L_{x_2}(x_2) \cdots L_{x_n}(x_n)$$

Separable filtering has the following advantages :

- reduction of the computing time of a convolution of dimension $\overbrace{p \cdots p}^{n}$ from p^n to np
- to take into account a noise having different characteristics along each direction (for instance in 3d medical images the noise can be different along directions X,Y and Z)
- generalization to any dimension
- use of recursive filtering (see next section)

The main drawback of separable filtering is that we can obtain anisotropic filters along directions not parallel to $x_1, x_2, \ldots, x_n$. Gaussian filters are among the very few isotropic separable filters that are isotropic.

Thus if L is a separable filter we obtain :

$$I \star L(x_1, \ldots, x_n) = I \star (L_{x_1}(x_1) \cdots L_{x_n}(x_n))$$

Therefore L can be implemented by the cascade of the filters : $L_{x_1}, \ldots, L_{x_n}$

If we suppose the noise homogeneous along any direction we can set :

$$L_{x_1} = L_{x_2} = \cdots = L_{x_n} = S$$

In the sequel we will suppose that the noise is isotropic but this method can be used for an anisotropic noise but homogeneous along directions $x_1, x_2, \ldots, x_n$. We will see that recursive filtering can be used only if the noise is homogeneous along each manifold of dimension 1 : $x_i = cste$, $i \neq j$. We obtain :

$$I \star L(x_1, \ldots, x_n) = I \star (S(x_1) \cdots S(x_n))$$

We set :

$$D = S' \; ; \; P = S''$$

It comes :

$$\frac{\partial I(x_1, \ldots, x_n)}{\partial x_i} = I \star (S(x_1) \cdots S(x_{i-1}) D(x_i) S(x_{i+1}) \cdots S(x_n))$$

$$\frac{\partial^2 I(x_1, \ldots, x_n)}{\partial {x_i}^2} = I \star (S(x_1) \cdots S(x_{i-1}) P(x_i) S(x_{i+1}) \cdots S(x_n))$$

We set :

$$Q_i(x_1, \ldots, x_n) = I \star (S(x_1) \cdots S(x_{i-1}) P(x_i) S(x_{i+1}) \cdots S(x_n))$$

It comes :

$$Laplacian(I) = I \star (Q_1(x_1, \ldots, x_n) + \cdots + Q_n(x_1, \ldots, x_n))$$

Therefore we have reduced the smoothing or the derivation of a signal of any dimension to the 1d case.

3 How to smooth or to derive a 1d signal

In this section we present some optimal [1] and recursive filters to smooth or to derive one or two times a 1d signal. We strongly insist on recursive implementation because of the low computational cost it allows [2]. For more details concerning these filters one can refer to [2].

3.1 Shen filter

$$s_1(x) = ce^{-\alpha.|x|}$$

c is chosen to obtain a normalized filter ie :

$$\int_{-\infty}^{+\infty} s_1(x)dx = 1$$

s_1 is a first order recursive filter having the following realization [12] :

$$\begin{array}{rcll} y^+(m) = & ax(m) + by^+(m-1) & \text{for } m = 1, \ldots, N \\ y^-(m) = & abx(m+1) + by^-(m+1) & \text{for } m = N, \ldots, 1 \\ y(m) = & y^-(m) + y^+(m) & \text{for } m = 1, \ldots, N \end{array}$$

$$s_1'(x) = \begin{cases} c.e^{-\alpha|x|} & \text{si } x \geq 0 \\ -c.e^{-\alpha|x|} & \text{si } x \leq 0 \end{cases}$$

$$a = c = \frac{1 - e^{-\alpha}}{1 + e^{-\alpha}} \; ; \; b = e^{-\alpha}$$

s_1' is a first order recursive filter having the following realization :

$$\begin{aligned} y^+(m) &= ax(m) + by^+(m-1) && \text{for } m = 1, \ldots, N \\ y^-(m) &= -ax(m) + by^-(m+1) && \text{for } m = N, \ldots, 1 \\ y(m) &= y^-(m) + y^+(m) && \text{for } m = 1, \ldots, N \end{aligned}$$

$$a = c = 1 - e^{-\alpha} \; ; \; b = e^{-\alpha}$$

This filter has been introduced by Shen and Castan to approximate the laplacian by difference between the original image and the smoothed image [12]. The derivation filter is an optimal solution for the first part of Canny criteria [12]. It corresponds to an optimal value for the product $\Sigma\lambda$ ie to the best trade-off detection-localization. The discontinuity of order 1 at point 0 avoids to delocalize the edges in the smoothed image when α is small. However this discontinuity can induce multiple edges.

3.2 Deriche filter

$$s_2(x) = (c|x| + 1)e^{-\alpha|x|}$$

s_2 is a second order recursive filter having the following realization :

$$\begin{aligned} y^+(m) &= a_0x(m) + a_1x(m-1) - b_1y^+(m-1) - b_2y^+(m-2) && \text{for } m = 1, \ldots, N \\ y^-(m) &= a_2x(m+1) + a_3x(m+2) - b_1y^-(m+1) - b_2y^-(m+2) && \text{for } m = N, \ldots, 1 \\ y(m) &= y^-(m) + y^+(m) && \text{for } m = 1, \ldots, N \end{aligned}$$

$$a_0 = k \; ; \; a_1 = k(\alpha - 1)e^{-\alpha} \; ; \; a_2 = k(\alpha + 1)e^{-\alpha} \; ; \; a_3 = ke^{-2\alpha} \; ;$$

$$b_1 = -2e^{-\alpha} \; ; \; b_2 = e^{-2\alpha} \; ; \; k = \frac{(1 - e^{-\alpha})^2}{1 + 2\alpha e^{-\alpha} - e^{-2\alpha}}$$

$$s_2'(x) = -cxe^{-\alpha|x|}$$

s'_2 is a second order recursive filter having the following realization :

$$\begin{aligned} y^+(m) &= ax(m-1) - b_1 y^+(m-1) - b_2 y^+(m-2) \quad \text{for } m = 1, \ldots, N \\ y^-(m) &= ax(m+1) - b_1 y^-(m+1) - b_2 y^-(m+2) \quad \text{for } m = N, \ldots, 1 \\ y(m) &= y^-(m) + y^+(m) \quad \text{for } m = 1, \ldots, N \end{aligned}$$

$$a = -ce^{-\alpha} \; ; \; b_1 = -2e^{-\alpha} \; ; \; b_2 = e^{-2\alpha}$$

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

s''_2 is a second order recursive filter having the following realization :

$$\begin{aligned} y^+(m) &= a_0 x(m) + a_1 x(m-1) - b_1 y^+(m-1) - b_2 y^+(m-2) \quad \text{for } m = 1, \ldots, N \\ y^-(m) &= a_2 x(m+1) + a_3 x(m+2) - b_1 y^-(m+1) - b_2 y^-(m+2) \quad \text{for } m = N, \ldots, 1 \\ y(m) &= y^-(m) + y^+(m) \quad \text{for } m = 1, \ldots, N \end{aligned}$$

$$a_0 = 1 \; ; \; a_1 = -(1+k\alpha)e^{-\alpha} a_2 = (1-k\alpha)e^{-\alpha} a_3 = -e^{-2\alpha} \; ;$$

$$b_1 = -2e^{-\alpha} \; ; \; b_2 = e^{-2\alpha} \; ; \; k = \frac{1-e^{-2\alpha}}{2\alpha e^{-\alpha}}$$

This filter has been recently proposed by R. Deriche [3] and its derivative is an exact solution to Canny equation extended for infinite filters.

4 Algorithm in the 3d case

4.1 Filtering stage

This stage can be easily formalized for any dimension, thus we describe it for whatever dimension. The 3d version is straightforward.

4.1.1 Choice of a 1d smoothing operator : $s(x)$

We strongly recommand to choice a filter that can be implemented recursively, mainly because of the computing time [2]. We can for example choose one of the two filters precedently described :

$$s(x) = s_2(x) = (c|x| + 1)e^{-\alpha|x|} \text{ Deriche filter}$$

$$s(x) = s_1(x) = ae^{-\alpha|x|} \text{ Shen filter}$$

Theoretically the derivation filter $s_2'(x)$ is better with respect to Canny's multiple response criterion, but $s_1'(x)$ meets the best trade-off detection-localization. For small values of α the shape of $s_2'(x)$ induce some delocalization problems. This drawback is shared by any filter whose impulse response is continuous at point 0. Neverless, the results dœs not significantly differ on many kinds of images.

4.1.2 Choice of the kind of approach : Gradient or Laplacian

Generally the computation of the second derivative is more sensible to noise. On the other hand its computational cost is lower, because of the simplifications that occur when computing the impulse response of the filter by multiplication and addition of smoothing and second derivative operators. However in the case of noisy images it is generally useful to threshold the zero crossing provided by the filtering stage , and this requires to compute the gradient magnitude at each zero crossing. The localization of the edges provided by the two kinds of methods is theoretically and experimentally the same. It may be point out that Laplacian approach tends to curve the right angles.

Ones we have chosen the filter and the kind of approach, we have to compute Gradient or Laplacian.

Let $I(x_1, \ldots, x_n)$ be an image of dimension n.

Let $G(x_1, \ldots, x_n)$ be the gradient of I.

$$G(I) = (\frac{\partial I}{\partial x_1}, \ldots, \frac{\partial I}{\partial x_n})^t$$

The computation of the Gradient components $\frac{dI}{dx_i}$ is done by computing images (D_i) corresponding to the partial derivatives with respect to x_i as follow :

for $i = 1, \ldots, n$ do
$\left[\begin{array}{l} D_i = I \\ \text{for } j \in [1, \ldots, n] \backslash \{i\} \text{ do} \\ [D_i = D_i \star s(x_j) \\ D_i = D_i \star s'(x_i) \end{array}\right.$

For an image of size p (ie $d_1 \times \cdots \times d_n$) the computation of each gradient component needs to compute p convolutions per point. Then we obtain for the entire computation $p^2 . \prod_{i=1}^{p} d_i$ convolutions per point.

If we use a direct implementation of a convolution mask 1d of size k, we obtain the following complexity : $kp^2 \prod_{i=1}^{p} d_i$. A recursive filtering of order r allows to obtain a complexity of order : $rp^2 \prod_{i=1}^{p} d_i$.

For example in 2d case we obtain for Deriche filter 13 multiplications and 12 additions per point and this whatever be the value of α.

Laplacian can be computed by adding the second derivatives . The computation of the second derivative may be done with the algorithmic structure precedently described where the first derivative operator $s'(x)$ is replaced by the second derivative operator $s''(x)$. However for some cases calculus simplifications occur allowing to reduce the computation cost [2].

4.2 From gradient or laplacian to 3d edges

Althought the computation of gradient or laplacian is the essential part of edge detection this stage does not provide directly the edge points. For one or second derivative approach two complementary stages have to be done. In this section we describe these stages in the 3d case.

4.3 Gradient approach

4.3.1 Extraction of the local gradient extrema

This section deals with the extraction of local gradient magnitude extrema in the gradient direction. The principle of this method is to move along the normal to the contour (approximated by gradient) and to select points having the highest gradient magnitude. We notice that this stage could also be generalized to whatever dimension.

Let $I(x, y, z)$ be a 3d image

Let $G(x, y, z)$ be the gradient of I at point (x, y, z)

Let M be a point of I

Let $G(M)$ be the gradient at point M

Let d be a given distance (eg $d = 1$)

Let M_1 and M_2 be the two points of the straight line including M and whose direction is $G(\vec{M})$, located at a distance d of M ; M_1 is taken in the gradient sense and M_2 in the opposite sense.

$$M_1 = M + d\frac{G(\vec{M})}{\|G(M)\|} \; ; \; M_2 = M - d\frac{G(\vec{M})}{\|G(M)\|}$$

We approximate the gradient at points M_1 and M_2 thanks to a linear approximation using the neighbours. An example is precisely described in [8]. Point M is selected if $N(M) > N(M_2)$ and $N(M) \geq N(M_1)$. To choose the strict maximum in the gradient direction and large maximum in the opposite sense is equivalent to select edge points on the brighter side of the border.

4.3.2 Thresholding of the gradient extrema

In real images, due to the noise, many local gradient extrema does not correspond to true edge points. The classical way to get rid of these false edge points is to perform some thresholding by using the gradient magnitude [11]. A very popular algorithm consists in selecting the gradient extrema where gradient magnitude is greater than a given threshold.

Unfortunately in many cases it does not exist a threshold enabling to select true edge points without introducing noisy edge points. A way to come over this problem is the hysteresis thresholding introduced by Canny for 2D edge detection [1]. The idea is to add the topological property of a contour that is to be a chain of connected edge points. We have extended this method to 3D and slightly improve it by adding a constraint (see later).

The principle of hysteresis thresholding is to select among all extrema whose gradient magnitude is higher than a low threshold t_l these such that it exists a connected path of extrema whose gradient magnitude is higher than t_l (i.e. connex component, that is a topological property) between the involved point and an extremum whose gradient magnitude is higher than a high threshold t_h. Moreover we can deal only with connected components which length is more than a minimal length $l_{\min}$. Then the following three properties will be verified in the edge map:

- the gradient magnitude of all selected extrema is higher than the low threshold t_l,
- these extrema are linked in connected components whose length is higher than $l_{\min}$,
- each of these connected component has at least an extrema whose gradient magnitude is higher that the high threshold t_h.

The algorithm can be split into three stages:

- determination of two images I_h and I_p such as:

$$\begin{aligned} I_h(M) &= 0 \text{ if } N(M) < s_h \ ; \\ I_h(M) &= 1 \text{ if } N(M) \geq s_h \ ; \\ I_l(M) &= 0 \text{ if } N(M) < s_l \ ; \\ I_l(M) &= 1 \text{ if } N(M) \geq s_l. \end{aligned}$$

  where $N(M)$ is the gradient magnitude at point M.

- expansion in connected components from all points such as $I_h(M) = 1$ on all points such as $I_p(M) = 1$. This stage consists in determining the adjacency graph of the points such as $I_p(M) = 1$ and then to select the connected components of the nodes such as $I_h(M) = 1$.

- selection of the connected components which length is higher than $l_{\min}$.

This thresholding algorithm can be improved in the following way.

The expansion in connected components is performed in any direction (by using 8-connexity in 2D or 26-connexity in 3D). But we have an estimation of the direction orthogonal to the contour i.e. the gradient direction. The idea is to move along a direction orthogonal to the gradient i.e. within the hyperplane tangent to the contour.

This may be done in the following way. Let M_0 be the edge point involved, $\vec{G}(M_0)$ be its gradient, V be the set of its neighborhoods (in 26-connexity for instance). The expansion

is performed by examination of all points $M \in V$ such as the distance between M and the hyperplane tangent at point M_0 is less than a threshold s:

$$\frac{\mid \vec{MM_0} \times \vec{G}(M_0) \mid}{\|\vec{G}(M_0)\|} < s$$

The choice of s is related to the curvature of the contour we want to obtain. We notice that by choosing s enough high, the expansion in connected components is done for all neighborhoods of M_0.

This improvement is essentialy useful in the case where the images are very noisy. And unfortunately this is true for many medical 3D images.

Particularly it allows to push down the low threshold without introducing too much false edge points.

This thresholding strategy is particularly efficient in the 3D case because it enables to get good connected edge points. This is of great interest to regroup these points in order to built surfaces.

4.4 Laplacian approach

In the case of a second derivative approach we have computed the laplacian value at each point. We suppose that the edge points are located at the laplacian zero crossing. A last thresholding stage is also necessary to remove zero crossing whose gradient magnitude is too low.

Classicaly the laplacian zero crossing extraction is split into three stages :

-1- *Determination of a polarity image*

We compute an image I_p such as :

$$I_p(M) = 0 \text{ if } Laplacien(M) > 0 \; ; \; I_p(M) = 1 \text{ if } Laplacien(M) \leq 0$$

-2- *Detection of Laplacian zero crossing*

We compute an image I_z such as :

$$\begin{array}{ll} I_z(M) & = 1 \text{ if } M \text{ corresponds to a transition 0-1 in } I_p \\ I_z(M-1) & = 1 \text{ if } M \text{ corresponds to a transition 1-0 in } I_p \\ I_z(M) & = 0 \text{ otherwise} \end{array}$$

We notice that the choice of the localization of the zero crossing at the point where laplacian is positive or negative allows to put edge points in the darker or brighter region.

-3- *Thresholding of Laplacian zero crossing*

We have chosen to use the hysteresis thresholding algorithm previously described.

5 Closing 3d edges

It is often very difficult to select adequate thresholds for the thresholding stage. High thresholds allows to remove noisy edge points but also true edge points ; low thresholds

allow to obtain all true edge points but also noisy edge points. We can not avoid this compromise and the choice of thresholds defines a trade-off between true and noisy edges.

Generally it is easier to extend uncompleted contours than to validate true contours in a noisy edge image. Thus we choose high thresholds to remove false edge points and then we use a tracking/closing algorithm. This algorithm could also be used to close open contours.

We have extended to 3d a 2d edge closing method proposed by Deriche and Cocquerez [4]. This 2d method supposes that it is possible to recognize endpoints of contours by the examination of a neighboroud 3×3 of an edge point. Each kind of neighborhoud is attached to a labelling code V. All possible configurations of contour endpoints are indexed. Moreover the topology of each endpoint allows to define an exploration direction to continue the contour. To each configuration is attached a list of neighbours to examine (see figure 1). The selected neighbours belong to the local gradient magnitude extrema points.

The implementation of this algorithm is easy. The image is scanned, if an edge point is identified as an extremity (ie the extremity configuration code is indexed) the algorithm is applied recursively to the involved extremity until a stop condition is verified.

Two choices have to be done :

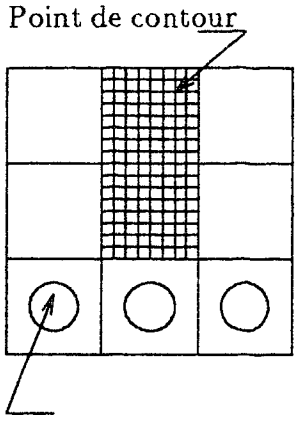
- *Choice of the neighboroud to continue the contour* : Deriche and Cocquerez select the point having the higher gradient magnitude. This works if the image is not too much noisy else we take interest in adding the condition that the candidate points have to belong to the gradient extrema image(see precedent section).

- *Stop conditions* : Many stop conditions may be used. Either there is no more candidates (if these ones are picked in the extrema image), either the path created recursively by the algorithm has reached an edge point or a given length.

The 3d extention of this algorithm is done by applying it on each plane XY, YZ, ZX and by adding the three edge images obtained. This can be justified by the assumption that the intersection of a 3d surface by at least one plane among XY,YZ and ZX is a curve on which the algorithm can be applied.

Despite of the result improvement due to the use of this tracking/closing algorithm, it remains some localized information lacks that cause holes of width 1 along X,Y or Z. These blanks are either due to the filter (a linear filter has not a good behavior near the multiple contours) either to the noise present in the image. To solve this problem we pick up the idea of the precedent algorithm, ie to attach a code to the neighbourhood of a point. Thus we select the codes corresponding to holes of width 1.

The 2d implementation consists in scanning the image and to fill up each identified hole. The 3d extension is the same than precedently (see figure 2).

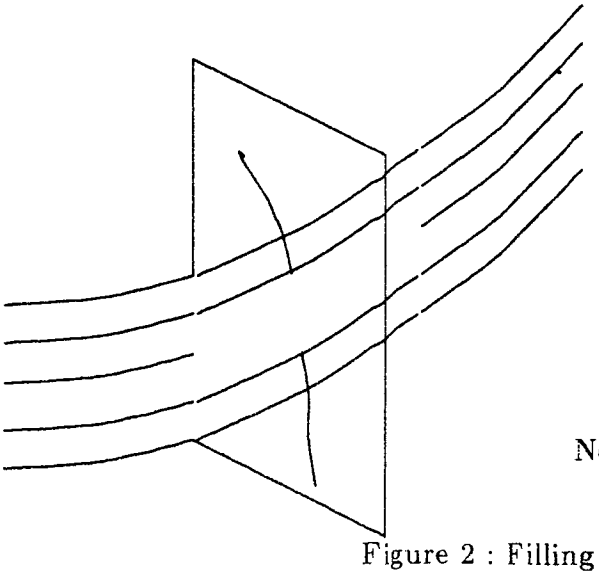


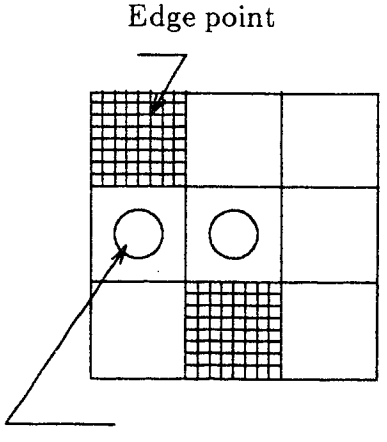
Neighbour to examine for closing : configuration code V=4

8	4	2
16		1
32	64	128

Values of neighbours to compute Code V

Figure 1 : Closing codes





Neighbour to examine for closing Code V=72

Figure 2 : Filling up holes

6 Experimental results

These algorithms have been implemented and tested on 3d magnetic resonance images of heart given by Hopital Hôpital Kremlin Bicêtre Paris. To obtain isotropic data (ie having the same resolution along X,Y,Z), we perform a linear interpolation along Z-axis. We suppose that the noise has the same parameters along X,Y,Z and thus we use the same operator width α to smooth or to derive in any direction.

Figures 3 to 12 show results obtained on a 3d NMR image of the left ventricle of size $64 \times 64 \times 46$ ($64 \times 64 \times 16$ before interpolation) and of resolution 1.5 mm. Figures 3 to 6 illustrate the interest in performing an hysteresis thresholding using the gradient direction (see section 4.3.2). We see that this allows to remove false edge points close to the true contour. Figures 7 to 12 show the main stages of our method. In figures 10 to 12 we see that the tracking/closing method enables to remove many spurious edge points. And this is done without spoiling a lot the computing time. Then the contour of the left ventricle is obtained and this without introducing any a priori knowledge.

Figures 13 to 12 present results provided by the same chain of processes but on a 3d NMR image of the chest ($256 \times 256 \times 46$,$64 \times 64 \times 16$ before interpolation, resolution 1.5 mm). We notice that the results obtained by Deriche and Shen filters cannot be compared because of the different meaning of α for the two operators. However other experimentations have shown that for the values of α used for such data ($0.5 < \alpha < 1$) the two family of filters give almost the same edge map and this for first and second derivative approach.

Figures 24 and 25 illustrate the interest of doing 3d edge detection instead of 2d edge detection. Figure 24 shows a cross section of the $3D$ edge image obtained on a cutting plane parallel to XZ. Figure 25 presents the same cross section but corresponding to the $3D$ image formed by the result of a $2D$ edge detection performed on the planes XY with the equivalent $2D$ algorithm. This example illustrate that $3D$ edge detection provides edges having a better correlation along the Z-axis. This is of great interest for all $3D$ treatment to be applied: surface reconstruction, $3D$ feature extraction$\cdots$.

We notice that for chest images where noise is important (particularly in the heart area) first derivative approach seems better.

For an experimental comparison of first derivative approach using Deriche filter with classical methods, one can refer to [8].

These algorithms has been implemented on a SUN 3 workstation. For a $256 \times 256 \times 46$ image, the filtering stage and the tracking/closing stage take respectively about 30 and 5' CPU.

7 Conclusion

We have proposed a 3d edge detection scheme that can be split into two main stages :

- Filering
- Edge tracking/closing

The key point of the filtering stage is to use optimal recursive and separable filters [2] to approximate gradient or laplacian. The recursive nature of the operators enables to implement infinite 3d impulse response with a computing time roughly similar to a $3 \times 3 \times 3$ convolution mask. This saving in computational effort is of great interest for 3d edge detection due to the huge size of 3d images. Then we obtain an algorithmic framework for the 3d edge detection filtering stage that combines better theoretical and experimental performances and a lower algorithmic complexity than the classical ones. Moreover, we stress that our approach can be used for any dimension.

The principle of the edge tracking/closing is to select from the previous stage, only the more reliable edge points and then to apply an edge closing method derived from the idea developed in [4]. This enables to improve substancially the results provided by the filtering stage. We point out that our tracking/closing algorithm is limited mainly because it is not really 3d.

Currently we investigate true 3d closing edge methods using discrete topology and mathematical morphology.

8 Acknowledgement

The authors would like to thank Dr. J.M. Rocchisani from Hôpital Cochin Paris and Dr. J. Bittoun from Hôpital Kremlin Bicêtre Paris for having provided the RMN images.

Références

[1] J.F. Canny. *Finding edges and lines in images.* Technical Report TR. 720, MIT, June 1983.

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

[3] R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision*, 1(2), May 1987.

[4] R. Deriche and J.P. Cocquerez. An efficient method to build early image description. In *International Conference on Pattern Recognition*, Rome, 1988.

[5] R. Gordon, G.T. Herman, and S.A. Johnson. Image reconstruction from projections. *Sci. Amer.*, 233:56–68, 1975.

[6] H.K. Lui. Two and three dimensional boundary detection. In *Comput. Graphics Image Process.*, pages 123–134, 1977. Vol. 6.

[7] O. Monga. An optimal region growing algorithm for image segmentation. In *International Journal of Pattern Recognition and Artificial Intelligence*, pages 351–376, December 1987.

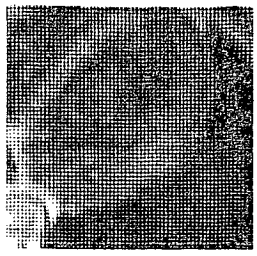


Figure 3 : Original cross section of NMR heart image corresponding to left ventricle

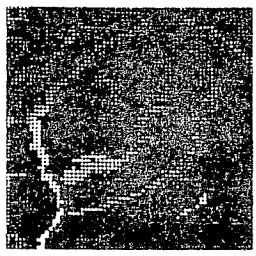


Figure 4 : Gradient magnitude image of the local gradient extrema

[8] O. Monga and R. Deriche. 3d edge detection using recursive filtering. In *Computer Vision and Pattern Recognition*, IEEE, San Diego, Juin 1989.

[9] O. Monga and R. Deriche. A new three dimensional boundary detection. In *International Conference on Pattern Recognition*, Paris, 1986.

[10] M. Morgenthaler and A. Rosenfeld. Multidimensional edge detection by hypersurface fitting. *PAMI-3*, 4, July 1981.

[11] A. Rosenfeld and A. Kak. Digital image processing. *New York: Academic*, 1976.

[12] J. Shen and S. Castan. An optimal linear operator for edge detection. In *Conference on Vision and Pattern Recognition*, IEEE, USA, Juin 1986.

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

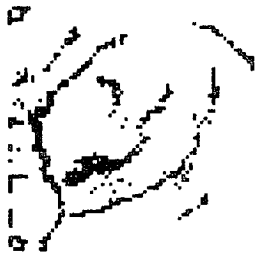


Figure 5 : 3d edges extracted thanks to hysteresis thresholding without constraint

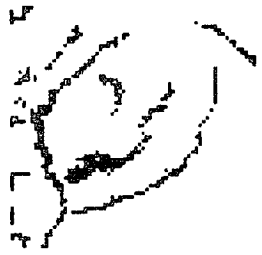


Figure 6 : 3d edges extracted thanks to hysteresis thresholding in the direction of the tangent plane

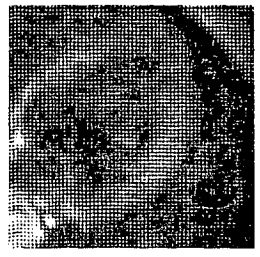


Figure 7 : Original cross section of NMR heart image corresponding to left ventricle

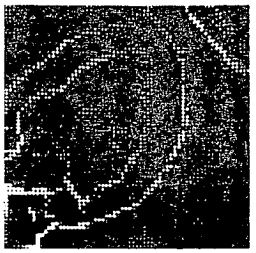


Figure 8 : Extrema image corresponding to precedent figure

Figure 9 : 3d edges obtained after hysteresis thresholding with high thresholds

Figure 10 : 3d edges after tracking/closing

Figure 11 : 3d edges after filling up holes of width 1

Figure 12 : 3d edges obtained after a simple hysteresis thresholding on the extrema image

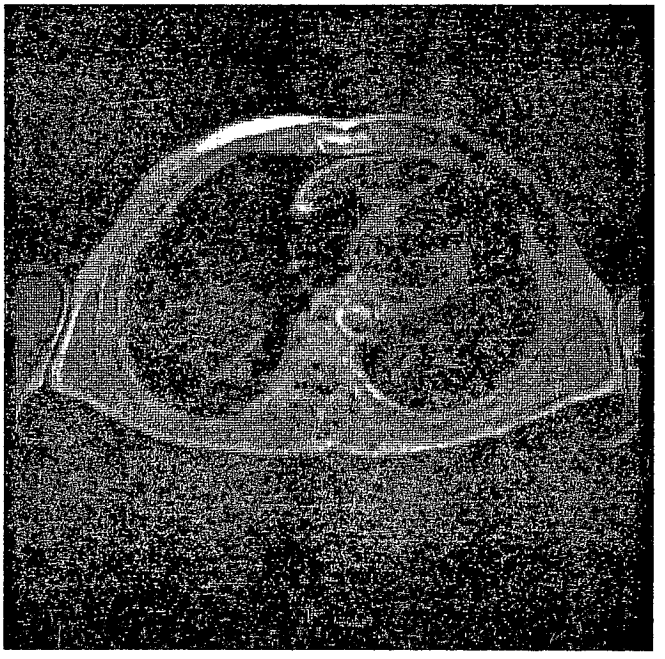


Figure 13 : Original cross section of RMN image corresponding to the diastolic cardiac phase

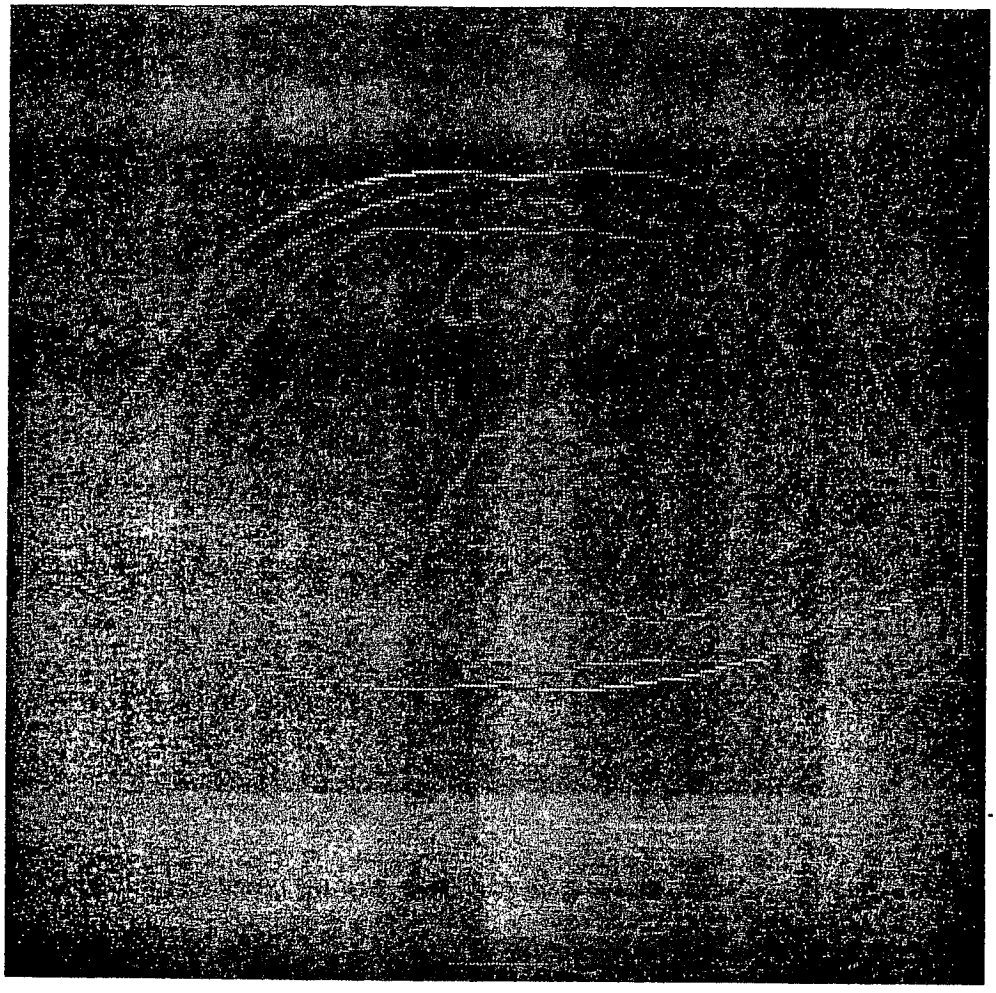


Figure 14 : Local 3d gradient extrema, Deriche filter, $\alpha = 0.6$

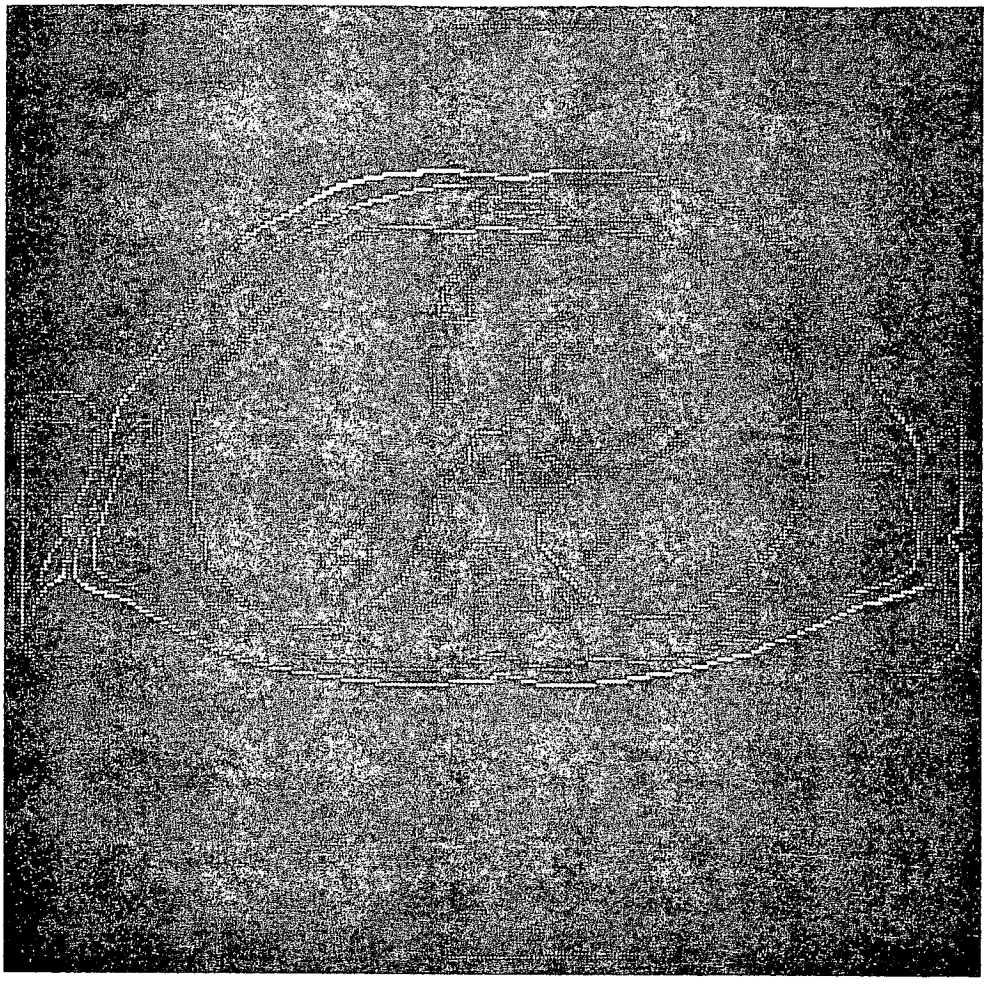


Figure 15 : Local 3d gradient extrema, Shen filter, $\alpha = 0.6$

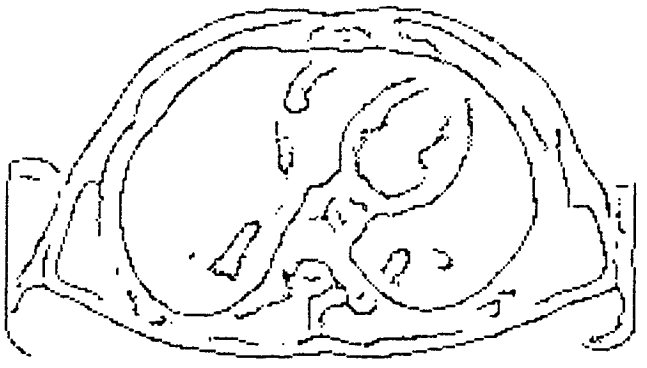


Figure 16 : 3d edges after hysteresis thresholding, Deriche filter, $\alpha = 0.6$

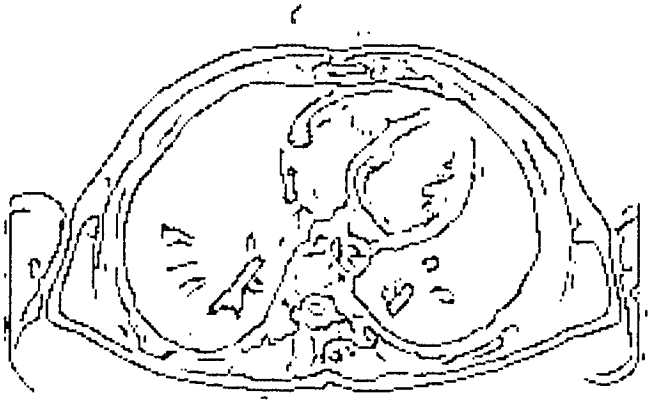


Figure 17 : 3d edges after hysteresis thresholding, Shen filter, $\alpha = 0.6$

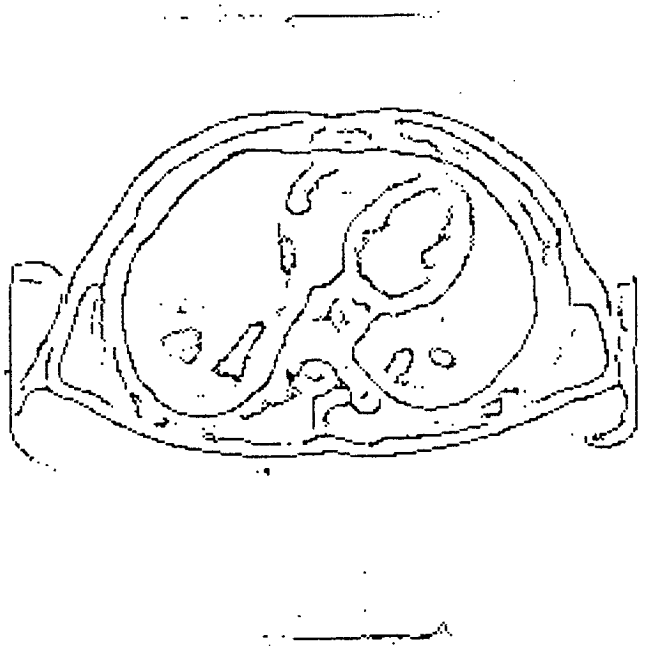


Figure 18 : 3d edges after tracking/closing, Deriche filter, $\alpha = 0.6$

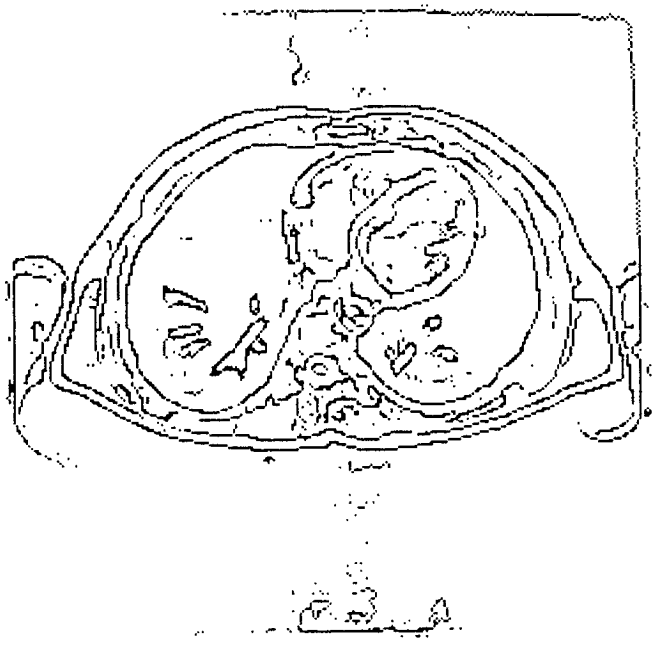


Figure 19 : 3d edges after tracking/closing, Shen filter, $\alpha = 0.6$

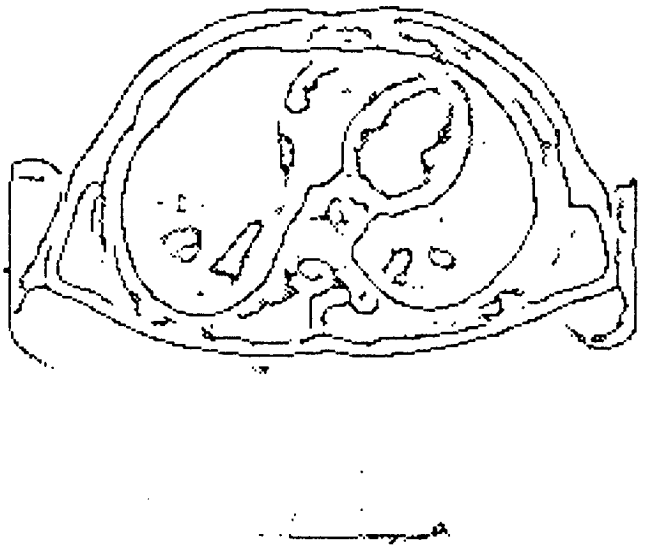


Figure 20 : 3d edges after tracking/closing and filling up holes of width 1, Deriche filter, $\alpha = 0.6$

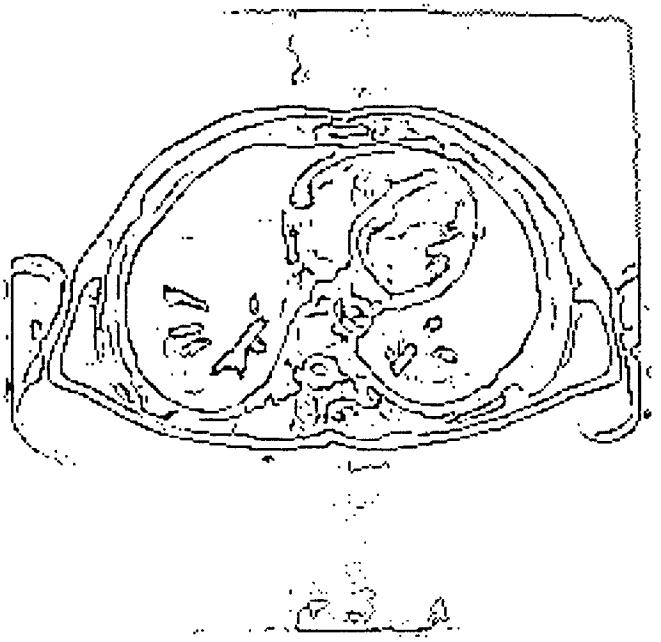


Figure 21 : 3d edges after tracking/closing and filling up holes of width 1 Shen filter, $\alpha = 0.6$

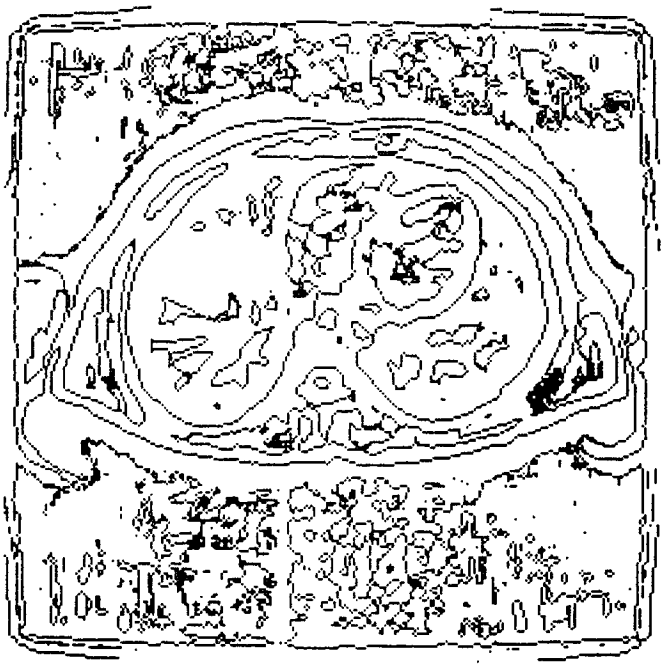


Figure 22 : 3d zero crossing without thresholding, Deriche filter, $\alpha = 0.6$

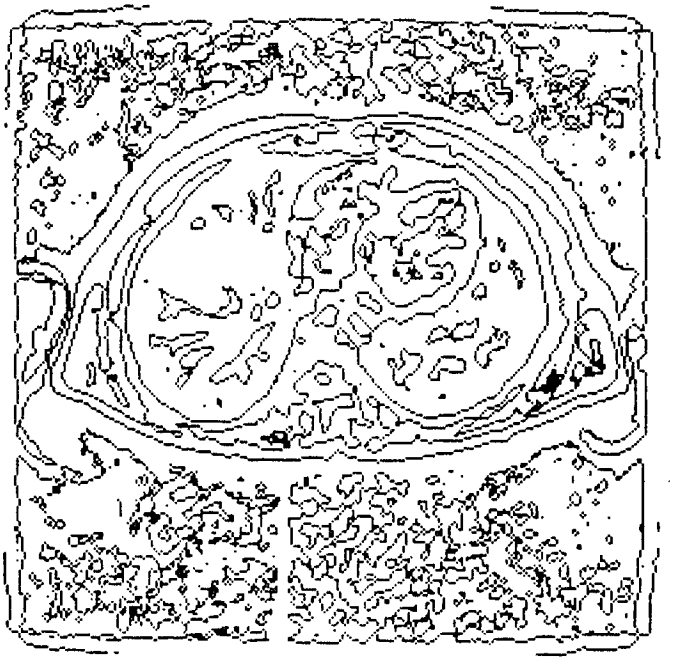


Figure 23 : 3d zero crossing without thesholding, Shen filter, $\alpha = 0.6$

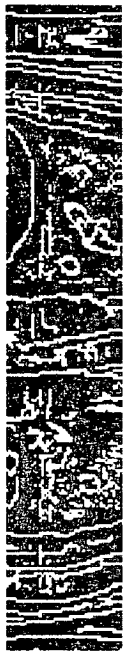


Figure 24 : Cross section of 3d image obtained by 3d edge detection corresponding to XY slice 128 at cardiac time 1

Figure 25 : Cross section of 3d edge map obtained by 2d edge detection on planes XY corresponding to XZ slice 128 at cardiac time 1

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