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 tracking/closing enables to extract many edges not provided by the filtering stage without introducing noisy edges. Experimental results obtained on NMR images are shown.

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 tracking/closing using morphological properties.

Then a first stage to identify these surfaces is to search their points using only signal information. This sets 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. Therefore, there is a trade-off to meet between the size of the convolution masks and their detection and localization performances. Particularly in the 3D case the huge size of data makes the algorithmic complexity and the storage requirement be key points.

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 zero crossings of the laplacian. 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. For this design we propose a 3D edge tracking/closing algorithm derived from a 2D edge closing method [4].

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 edge 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

Classically edge detection is tackled thanks to the following two schemes :

1. Gradient computation and extraction of the local extrema of its magnitude in the gradient direction : gradient approach.

2. Laplacian computation and determination of its zero crossings : Laplacian approach

Most of the techniques use linear filters to approximate the gradient or the laplacian. Given that the derivative of a convolution product is equal to the convolution of the signal by the derivative of the filter response, the schemes 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 computation of the laplacian 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 sets 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 filters having separable response with respect to directions $x_1, x_2, \ldots, x_n$:

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

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 separable filters which are isotropic.

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 : $x_i = cste,\ i \neq j$.

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

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

s_1 and s_1' are first order recursive filters. s_1 has the following realization [12] :

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

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 derivative 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, s_2', and s''_2 are second order recursive filters [3].

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 any dimension.

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_1 or s_2.

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 do 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 experimentally the same. It may be point out that Laplacian approach tends to smooth 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 :

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

For an image of size p (ie $d_1 \times \cdots \times d_p$) 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.

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 the 2D case we obtain for Deriche filter 13 multiplications and 12 additions per point and this for any value of α.

The 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 computational cost [2].

4.2 From gradient or laplacian to 3D edges

Although the computation of gradient or laplacian is the essential part of edge detection it does not provide directly the edge points. For first 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 any 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]. The 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

We have extended the hysteresis thresholding introduced by Canny for 2D edge detection [1] to 3D and slightly improved it by adding a constraint. 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 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 whose the length is more than a minimal length $l_{\min}$.

This thresholding algorithm can be improved in the following way. The expansion in connected components could be performed in any direction (by using 8-connectivity in 2D or 26-connectivity 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 neighbourhoods (in 26-connectivity for instance). The expansion is performed by examination of all points $M \in V$ such that the distance between M and the hyperplane tangent at point M_0 is less than a threshold s:

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

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 neighbourhoods of M_0.

This improvement is essentially 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 [11]. A last thresholding stage is also necessary to remove zero crossing whose gradient magnitude is too low.

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.

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 neighbourhood 3×3 of an edge point. Each kind of neighbourhood is attached to a labeling code. 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. 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 neighbourhood 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.

- *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 extension 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

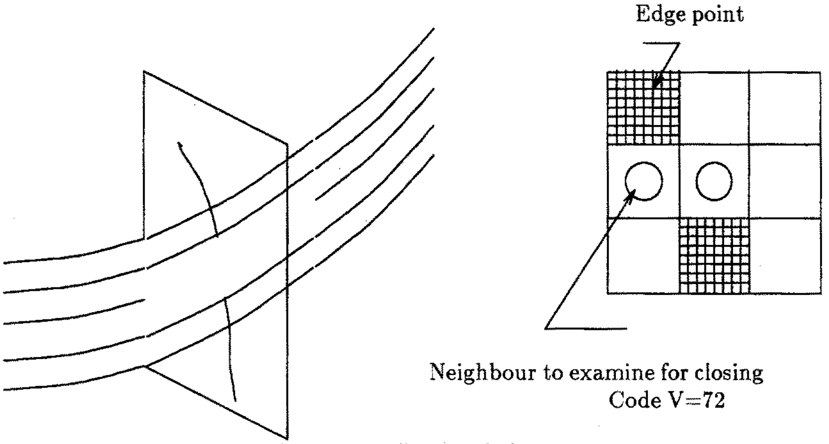

Figure 1: Filling up holes

holes of width 1 (see figure 1). The 2D implementation consists in scanning the image and to fill up each identified hole. The 3D extension is the same than precedently.

6 Experimental results

These algorithms have been implemented and tested on 3D magnetic resonance images of heart given by 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 2 to 5 present results provided by the chain of processes on a 3D NMR image of the chest ($256 \times 256 \times 46$, $256 \times 256 \times 16$ before interpolation, resolution 1.5 mm).

These algorithms have 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' CPU and 5' CPU.

7 Conclusion

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

- Filtering
- 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 could be used for any dimension.

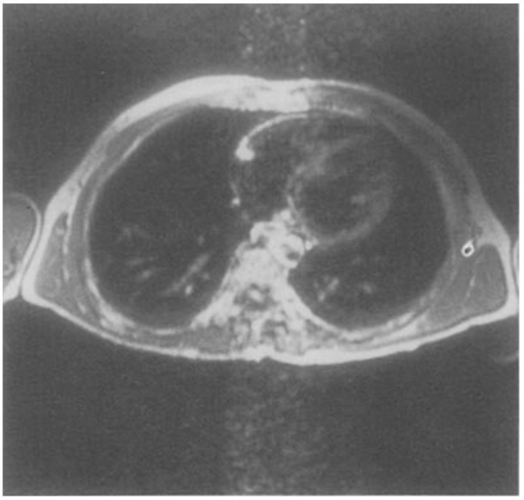

Figure 2: Original cross section of a RMN image of the chest corresponding to the diastolic cardiac phase

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.

References

[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.

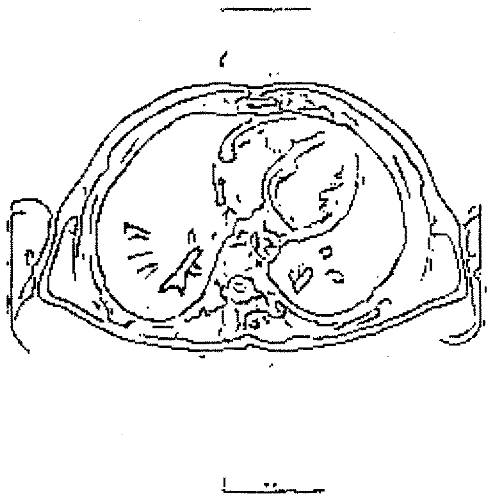

Figure 3: 3d edges after hysteresis thresholding, Shen filter, $\alpha = 0.6$

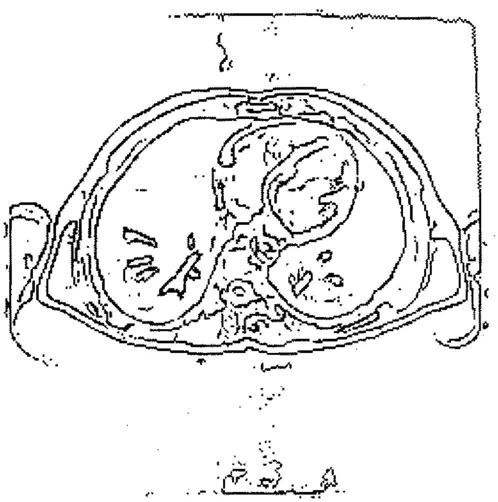

Figure 4: 3d edges after tracking/closing

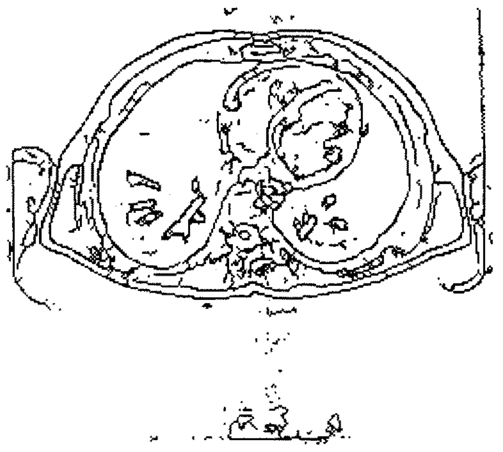

Figure 5: 3d edges after tracking/closing and filling up holes of width 1

[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.

[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.