3D Contour Closing: A local operator based on Chamfer distance transformation

Minh-Phuong Tran

3D Contour Closing: A local operator based on Chamfer distance transformation
Apply to biological image of mice vessels network

Minh-Phuong Tran*

March 18, 2013

Abstract

Visual perception evolved in a world of objects many of which are bounded by smooth closed contours. Especially with the three-dimensional dataset that need to be marked the locations of filaments to get the region of interest. There also exist other structures with high values (noise) or we may loose some information within the image through contour detection process. For instance, a simple global thresholding method based on gradient edge detection will give us poor results even when if we choose an optimal threshold value. Indeed, there exist many other structures with high grey level values. Global thresholding may keep noise if threshold parameter is too low, and we may loose some image information if threshold parameter is too large.

The paper describes a new contour closing method, a local operator based on the so-called Chamfer distance transformation. Inside we give a quick overview of the original philosophy and motivation of this algorithm. In addition, some numerical simulations will be presented and applied to the 3D biological mice vessel network.

1 Introduction

Contour detection strategy plays an important role in the computer vision community, it is driven by the structure of images. Especially, in the case of three dimensional medical images, a local context of a contour significantly affects the global saliency of the contours.

Classically, there are a lot of known contour detection methods that have been developed during the past decade. A simple global thresholding, based on gradient edge detection will give us poor results even when we choose an optimal threshold value. Indeed, there exist many other structures with high grey level values. Global thresholding may keep noise if threshold is too low, and we may loose some information if threshold is too large. That is the reason why we have to perform contour closing algorithms.

One of the simple closing algorithm is hysteresis thresholding method, that is usually adopted in the Canny detector. With hysteresis thresholding, the obtained object boundaries given are usually more complete and continuous then those given by pixel based techniques. However, it gives some false features of detected image. In particular, two

*Department of Mathematics and Applications, École Normale Supérieure, 45 rue d'Ulm - F 75230 Paris cedex 05, France (Minh-Phuong.Tran@ens.fr)

threshold values determination creates a lot of false pixels around objects contour shapes. We cite [7] for further results and examples.

We now study another local operator that has been proposed to close contours. The method is described in [5]. The "chamfer" distance measures distance between elements. In what follows a *contrast point* is a point where the image intensity gradient is high. Contour points are contrast points if we precisely define what "high" means.

The basic idea of the method of [5] is the following scheme:

(a) Generate a distance Table where the intensity (grey level) of a point is replaced by the distance to nearest contrast point.

(b) Find the saddle-points of the intensity function.

(c) Eliminate the points which associated distance is too large;

(d) Add those saddle points to contrast points and repeat steps $(a), (b), (c)$ and (d) as far as possible;

(e) Remove the contour points that do not close any the "meaningful" area.

Therefore the first step is the contrast points extraction. In this section, the concept of distance transformation and some of their properties, applications are described. We start with the two dimensional distance transform (DT), and the generation of distance transform to higher dimensions (three dimensions in particular) will be given. The distance transform was described in [4] and we just give hereafter some discrete definitions and implementations of DT.

2 Image Distance Transform

2.1 Two dimensional Distance Transform

Definition 2.1. [5], [4] **Discrete Distance**
Let $\mathbb{E} = \mathbb{Z}^N$, $N \geq 2$. An image I is an application defined on $\mathbb{E}$.
We call $d : \mathbb{E} \times \mathbb{E} \to \mathbb{N}$ the discrete distance if it verifies: $\forall A, B, C \in \mathbb{E}$

1. *$d(A, B) \geq 0$; $d(A, B) = 0 \Leftrightarrow A = B$: positive;*
2. *$d(A, B) = d(B, A)$: symmetric;*
3. *$d(A, B) \leq d(A, C) + d(C, B)$: triangular inequality.*

Definition 2.2. [5], [4] **Distance Transformation**
Let d be a distance on $\mathbb{R}^2$, I be 2D binary image and $O \subset I$. For each pixel in the background $\bar{O} := I \backslash O$, the distance transformation image denoted DT is defined as following:

$$DT : p \mapsto DT(p) = \min \left\{ d(p, q), q \in \bar{O} \right\} .$$

In other terms, the value assigned at any point p is the shortest distance from p to all points in O.

(a) Telephone - Original image (b) Telephone - Binary image (c) Telephone - Distance image (Euclidean)

Figure 1: A 2D example for distance transformation.

Depending upon various applications, the distance function may be defined in different ways, some most important kinds of function that are included in [4]. For example, the n-neighbor distances, they are 4- and 8-neighbor distance (for two dimensional case) and $6-$, $18-$ and $26-$neighbor distance (for three dimensional case), the Euclidean distance, octagonal distance, and the chamfer distance ... Throughout this paper, we are interested in choosing of chamfer distances map for a binary image respectively.

In two dimensions, each pixel has two kinds of neighbors. The first kind is 4 horizontal/vertical neighbors, the neighbors joined by a line. The second kind is the 4 diagonal neighbors, the neighbors joined only by a point. The global distance is computed from the local distances between neighbors. The distance between horizontal/vertical neighbors is denoted d_1, and the distance between diagonal neighbors is denoted d_2.

For example, in [4], with 4-neighbors distance we have $d_1 = 1$ and $d_2 = \infty$ (this means that all sums including d_2 can be ignored), the discs of this distance are diamonds; for 8-neighbors distance $d_1 = 1$ and $d_2 = 1$, the discs of this distance are squares.

The (2D) chamfer distance transform has real values based on d_1 (horizontal) and d_2 (vertical) as well. The name of this distance transform comes from the two-pass process for obtaining the distances, which is known as *chamfering*. However, in most applications real numbers are not desirable. The good integer approximations of the optimal values are found, $d_1 = 2$, $d_2 = 3$, the incremental distance values of 2 and 3 provide relative distances that approximate the Euclidean distances 1 and the square root of 2. This is why it is an euclidian pseudo-distance.

Position	$j-1$	j	$j+1$
$i-1$	$+d_2$	$+d_1$	$+d_2$
i	$+d_1$	0	

	0	$+d_1$	i
$+d_2$	$+d_1$	$+d_2$	$i+1$
$j-1$	j	$j+1$	Position

Table 1: The forward pass (left) and backward pass (right) from $I(i, j)$, by chamfer algorithm.

The algorithm for distance transformation can be described by masks in Table 1. Inside these masks, two passes over the picture are necessary:

- During the forward pass the mask starts at the upper left corner of the picture, moves row-by-row to perform computation on image I from left to right, and from top to bottom to find the partial distance value $d(i, j)$ for every pixel (i, j). The

local distances, d_1 and d_2, in the maskpixels are added to the pixel values in I, and new value of the zero pixel is the minimum of five sums. During the forward pass, the optimal distance for the pixel (i, j) of the mask is updated by the formula:

$$d(i,j) = \min \left\{ \begin{array}{l} d(i,j); \\ d(i,j-1) + d_1; \\ d(i-1,j-1) + d_2; \\ d(i-1,j) + d_1; \\ d(i-1,j+1) + d_2 \end{array} \right\}. \tag{2.1}$$

- Similarly, in the backward pass the mask starts in the lower right corner, executes from right to left, and from bottom to top to find the second partial distance value $d(i, j)$ for each pixel (i, j). The distance for pixel (i, j) of the mask can be calculated in terms of:

$$d(i,j) = \min \left\{ \begin{array}{l} d(i,j); \\ d(i,j+1) + d_1; \\ d(i+1,j-1) + d_2; \\ d(i+1,j) + d_1; \\ d(i+1,j+1) + d_2 \end{array} \right\}. \tag{2.2}$$

Table 2 gives an example, shows the results of different settings of 2D distance transformation for a certain 5×5 image where the black pixel is the object.

	■			

3	2	3	4	5
2	1	2	3	4
1	0	1	2	3
2	1	2	3	4
3	2	3	4	5

2	2	2	2	3
1	1	1	2	3
1	0	1	2	3
1	1	1	2	3
2	2	2	2	3

5	4	5	6	8
3	2	3	5	7
2	0	2	4	6
3	2	3	5	7
5	4	5	6	8

Table 2: Results of different settings distance transformations. From left to right: the first Table is the 5×5 image with black pixel is an object; the second one is the result after 4-neighbor distance; next after 8-neighbor distance and the last one is result after chamfer distance transformation.

2.2 Three dimensional Distance Transform

Three-dimensional elements that are corresponding to two-dimensional case, are usually called voxels, and there are $6-$, $18-$ or $26-$neighbor distances may be considered. For instance, with $26-$neighbor distance, voxel has 26 neighbors of three different kinds. The first, closest, kind of neighbors is the 6 ones joined to the voxel by a plane, the second kind is the 12 neighbors joined by a line, and the third kind is the 8 neighbors joined by only a point. The three different local distances are denoted d_1, d_2 and d_3. As in two dimensions, the algorithm can be illustrated by the same masks as in Table 3 by using d_1, d_2 and d_3, and two passes over the volume are needed. The forward mask is moved over the volume left to right, top to bottom, and front to back. The backward mask is moved in the opposite way. In each position, the sum of the local distance in each maskvoxel and

$+d_3$	$+d_2$	$+d_3$
$+d_2$	$+d_1$	$+d_2$
$+d_3$	$+d_2$	$+d_3$

$+d_2$	$+d_3$	$+d_3$
$+d_1$	0	

	0	$+d_1$
$+d_2$	$+d_1$	$+d_2$

$+d_3$	$+d_2$	$+d_3$
$+d_2$	$+d_1$	$+d_2$
$+d_3$	$+d_2$	$+d_3$

Table 3: The forward pass (first row) and backward pass (second row) in 3D case, by chamfer algorithm.

the value of the voxel it covers is computed, and the new value of the zero voxel is the minimum of these sums.

As remarked before in two-dimensions, the values of d_1, d_2 and d_3 are set dependently on which kind of distance transformation. For example, in [4] we have:

- $6-$neighbor distance: $d_1 = 1$, $d_2 = d_3 = \infty$.
- $18-$neighbor distance: $d_1 = 1$, $d_2 = 1$ and $d_3 = \infty$.
- $26-$neighbor distance: $d_1 = d_2 = d_3 = 1$.
- Chamfer distance: as 2D case, it is often desirable to use only integers. A very good integer approximation of the optimal local chamfer distances is $d_1 = 3, d_2 = 4$, and $d_3 = 5$. We can refer to [5] for a detailed explanation, and this chamfer $(3, 4, 5)$ distance is used in this chapter.

3 A local operator based on Chamfer Distance Transform

3.1 Saddle Points Determination

Definition 3.1. [2] *Let $f : V \times W \to \mathbb{R} \cup \{\infty\}$ be a function defined on $V \times W$, where V and W are vector spaces. A point $(a, b) \in V \times W$ is called a saddle point of f if:*

$$\begin{cases} a \in \arg\min\limits_{x \in V} f(x, y) \\ b \in \arg\max\limits_{y \in W} f(x, y) \end{cases}$$

Proposition 3.2. [2] *The point (a, b) is a saddle point of f if and only if:*

$$\min_x \max_y f(x, y) = \max_y \min_x f(x, y) = f(a, b) \tag{3.1}$$

Proof. Recall the definition, $\forall (x_0, y_0)$ we have:

$$\max_y f(x_0, y) \geq f(x_0, y_0) \geq \min_x f(x, y_0) \tag{3.2}$$

We can conclude the proposition since the above inequality still holds when we move to min and max on the left and right, so the saddle point definition is equivalent to:

$$\max_y f(x_0, y) \geq f(x_0, y_0) \geq \min_x f(x, y_0) \tag{3.3}$$

□

We give thereafter equivalent formulations:

Definition 3.3. [5] *Let $f(x,y)$ be a real function of two variables defined over a real open subspace $\Omega \subset \mathbb{R}^2$ and differentiable with respect to these two variables on Ω. We say that the point $(a,b) \in \Omega$ is the saddle point of f if:*

(a) $\dfrac{\partial f}{\partial x}(a,b) = \dfrac{\partial f}{\partial y}(a,b) = 0,$

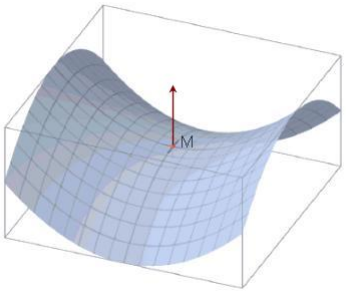
(b) *(a,b) is neither a local maximum nor a local minimum of f.*

Definition 3.4. [5] *Consider now an open bounded subset K of Ω (K will play the role of contour points in our saddle point method). Let a function $f_K : \Omega \to \mathbb{R}$ be defined as:*

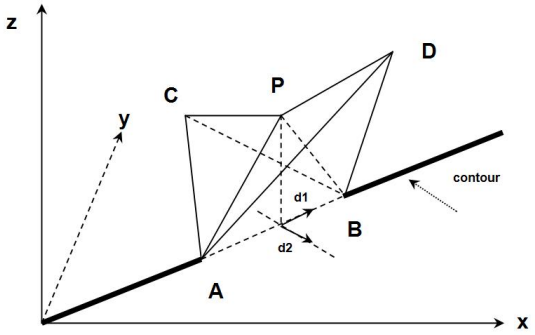
$$f_K(x,y) = \min\{d((x,y),(u,v)) \,|\, (u,v) \in K\} = d((x,y),K)$$

We say that $P \in \Omega$ is a saddle point if there exist two directions d_1 and d_2 in $\mathbb{R}^2$ such that $f(P)$ is the maximum in direction d_1 and the minimum in d_2 :

$$\exists d_1, d_2 \in \mathbb{R}^2, \exists r > 0, \forall \lambda : |\lambda| \leq r \Rightarrow \begin{cases} f(P + \lambda d_1) < f(P) \\ f(P + \lambda d_2) > f(P) \end{cases}$$



(a) Example for the 1^{st} and 2^{nd} definition of Saddle point, with $z = x^2 - y^2$, $M(0,0)$ is the saddle point.



(b) The 3^{rd} definition of Saddle point.

Figure 2: The saddle point example presentations.

Proposition 3.5. [2], [5]
Let K be a finite union of segments in space and A, B be two end (distinct) points of two segments in K. Let P be the medium of (A,B) and $r = d(A,P) = d(B,P)$. We denote $D(M,t)$ the closed disk with center M and radius t.
Suppose that we have:

(a) $D(P,t) \cap K = \{A,B\}$.

(b) $\exists \varepsilon > 0, D(P, r+\varepsilon) \cap K = \{(A,A'),(B,B')\}$.

Then, P is the saddle point of f_K, that means:

- *$f_K(P)$ is the local minimum respect to direction d_1.*
- *$f_K(P)$ is the local maximum respect to direction d_2.*

This proposition is well proved in [5].

3.2 Numerical Saddle Points Discretization

We now consider the 2D discrete case corresponding to a scanned image (the whole Ω) and a set of contour points (the points of contrast). In this case, we look for the discretized saddle points y using 8-neighbor pixels of pixel P.

The point P is called a (discretized) saddle point if there exists four points I, J, Q, R such that:

(1) I, J and P are aligned.

(2) Q, R and P are aligned.

(3) The segments (I, J) and (Q, R) intersect each other.

(4) $f_K(I) \leq f_K(P)$ and $f_K(J) \leq f_K(P)$.

(5) $f_K(Q) \geq f_K(P)$ and $f_K(R) \geq f_K(P)$.

(6) At least three of the inequalities represented by (4) and (5) are strict.

Let us give an example in Table 4:

							I			
$I \rightarrow$	1	4	2			3	3	3		
$Q \rightarrow$	4	3	5	$\rightarrow R$	$R \rightarrow$	4	4	5	$\rightarrow Q$	
	7	4	2	$\rightarrow J$		3	3	3		
							J			

Table 4: Some example of a 2D discretized saddle point.

3.3 Contour Closing Algorithm

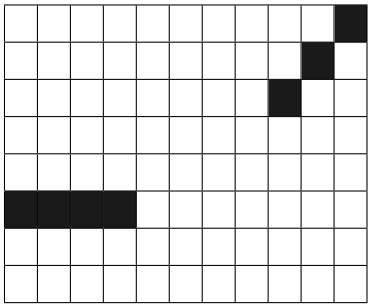
Let K_0 the set of all contrast points (closed contour positions) that need to be updated, and let f represents the Chamfer distance transform of our considered image, which is described in Definition 3.4.

We have $f_0(M) = f_{K_0}(M) = d(M, K_0)$ (we want to check if M is the saddle point or not).

1. Step 1: $K = K_0$, for initial set of saddle points.
2. Step 2: Compute $\mathcal{P}(f_K)$, is the set of all saddle points of image f_K.
3. Step 3: If $\mathcal{P}(f_K) = \emptyset$ then go to step 4. If not, update $K = K \cup \mathcal{P}(f_K)$ and repeat step 2.
4. Step 4: Stop.

The process of found and replaced saddle points by the contour positions is iterated until a contour is reached or until a fixed number of iterations is attained. Let consider an example in Table as following, which represents two segments. We obtain how this algorithm performs for closing them. For this simple example, we test with a 2D local

10×10 image, it can be supposed that by our detection method, the unclosed contours are presented as black pixels in Table 5 on left. Its chamfer distance transform could be calculated and presented on the right image, in which the image takes zero values at all our available contour positions need to be closed.



10	10	10	10	10	8	6	5	3	2	0
8	8	8	8	9	7	5	3	2	0	2
6	6	6	6	7	6	4	2	0	2	3
4	4	4	4	5	6	5	3	2	3	5
2	2	2	2	3	5	6	5	4	5	6
0	0	0	0	2	4	6	7	6	7	8
4	2	2	2	3	5	7	9	8	9	10
∞	4	4	4	5	6	8	10	10	11	12

Table 5: Left: Object that need to be closed with two black segments. Right: The chamfer distance transformation of the object.

10	10	10	10	10	8	6	5	3	2	0
8	8	8	8	9	7	5	3	2	0	2
6	6	6	6	7	6	4	2	0	2	3
4	4	4	4	5	6	0	3	2	3	5
2	2	2	2	3	0	6	5	4	5	6
0	0	0	0	2	4	6	7	6	7	8
4	2	2	2	3	5	7	9	8	9	10
∞	4	4	4	5	6	8	10	10	11	12

Table 6: Closing result of the object in Table 5 after one iteration, two saddle points are found and replaced.

10	10	10	10	10	8	6	5	3	2	0
8	8	8	8	9	7	5	3	2	0	2
6	6	6	6	5	3	2	0	0	2	3
4	4	4	4	3	2	0	0	2	3	5
2	2	2	2	0	0	2	3	4	5	6
0	0	0	0	0	2	3	5	6	7	8
4	2	2	2	3	4	5	6	8	9	10
∞	4	4	4	5	6	8	10	10	11	12

Table 7: Closing result of the object in Table 5 after two iterations, four next saddle points are found and replaced. At this time we have $\mathcal{P}(f_K) = \emptyset$, we stop.

4 Numerical Experiments

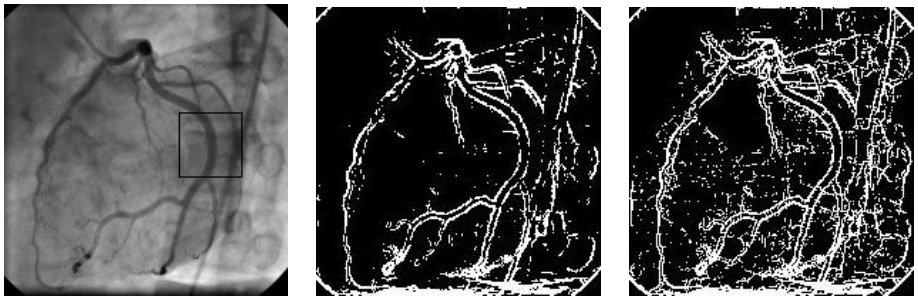
In this section, we consider that the object has been successfully segmented once we have a closed contour approaching this set of unconnected points. A small remark that our

detection method was chosen as a local histogram shape based thresholding, see more in [7].

The chamfer distance operator could be applied for closing contours after contour detection step. Moreover, this step-contour-detector is well-suited to extract contour shapes, especially our three-dimensional medical images consideration.

4.1 Test on 2D MRI scan

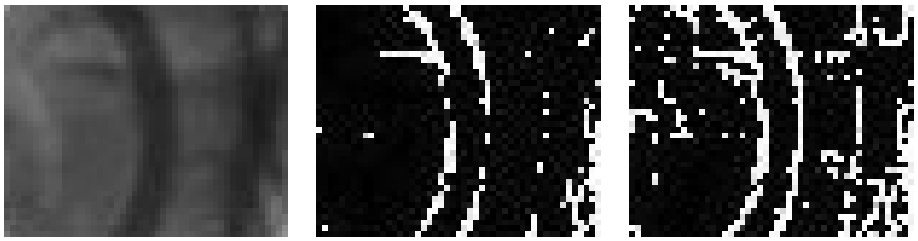
Here we apply our closing contour method and present numerical results tested for 2D MRI scan. This MRI scan models the eye-blood vessel. Test image is shown in figure 4.1(a), the zoom of image can also be considered, shown in Figure 4.1(d). On the second column of the same figure, we present detected contour applied thresholding step. Experimental results can be represented in the Figure 4.1(c) and (f). In the zoom figure of blood vessels 4.1(f), the efficiency has been also evaluated. In particular, a lot of the contour pixels which were not detected in segmented process are found. It requires little computation. The presence of saddle points is taken during a second scan of the image. The obtained results are good and allow the closure of majority of gaps of 5 and 10 pixels after a few iterations. This method can be used to perform preprocessing during image segmentation operations. It is more efficient than the hysteresis thresholding method, we can refer to [7] for more details.



(a) 2D example of original eye-blood vessel.

(b) Detected contour by a simple thresholding step.

(c) Detected contour found by Chamfer distance transform.



(d) Zoom of original blood-vessel.

(e) Zoom of detected contour by simple thresholding.

(f) Zoom of detected contour found by Chamfer distance transform.

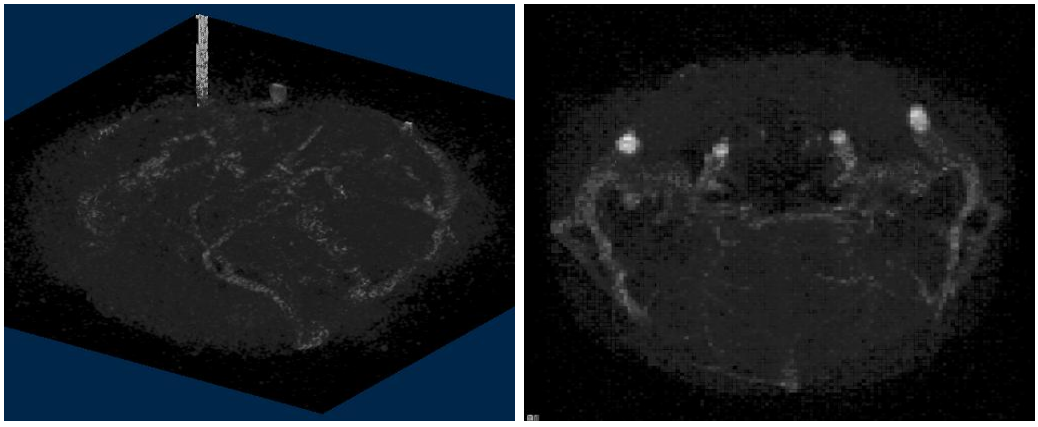
Figure 3: An example of 2D eye-blood-vessels test, contours found by chamfer distance transform closing method.

Remark 4.1. *Because of the discretization process, there are some cases with false detection. It is proposed to catch neighboring pixels that mark contours. Finally, we must perform a selection process of contour positions accurately to find all true contours. This problem can be resolved by making two types of filtering at each iterations.*

- *When these false detections occur for distant points of the contours, it can be carried out by setting a minimum distance of intervention of the operator;*
- *Assuming that the contours separate regions homogeneous grayscale, false saddle-points do not correspond to contrast points. Gradient variations at these points are low, this can be used to not validate them.*

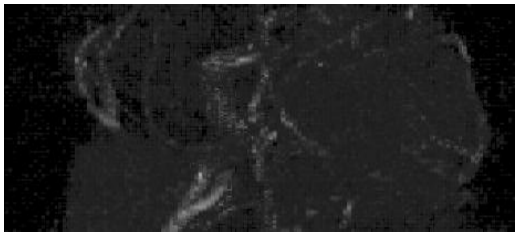
4.2 Test on 3D MRI biological mice vessels network

Let us first describe the three dimensional model of the brain vessels network. This volume taken from CBM Laboratory[1] in Orléans. The vessels of mice brain stack is composed of 51 two dimensional MRI slices. In this case, biologists want to recover the network of filament structures. The Figure 4 describes the whole 3D volume visualization.

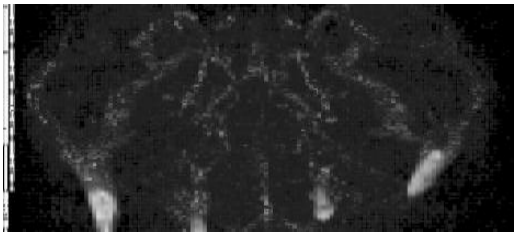


(a) 3D noisy Vessels data.

(b) Noisy volume, along xy cross direction.



(c) Noisy volume, along yz cross direction.

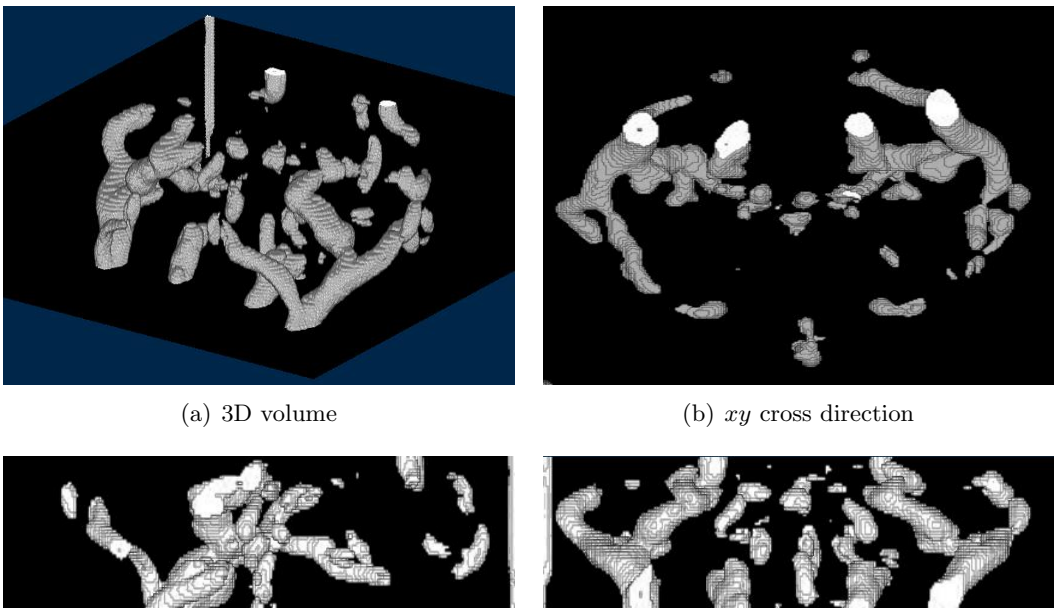


(d) Noisy volume, along xz cross direction.

Figure 4: Original 3D vessel volume.

In [7], we note that contours are well identified and a threshold parameter is used to obtain a binary segmented image. Binary results are shown in Figure 5 below, we can cite [7] for the applied contour detection methods.

[1]CNRS/CBM: Centre de biophysique moléculaire `http://cbm.cnrs-orleans.fr/`



(a) 3D volume

(b) xy cross direction

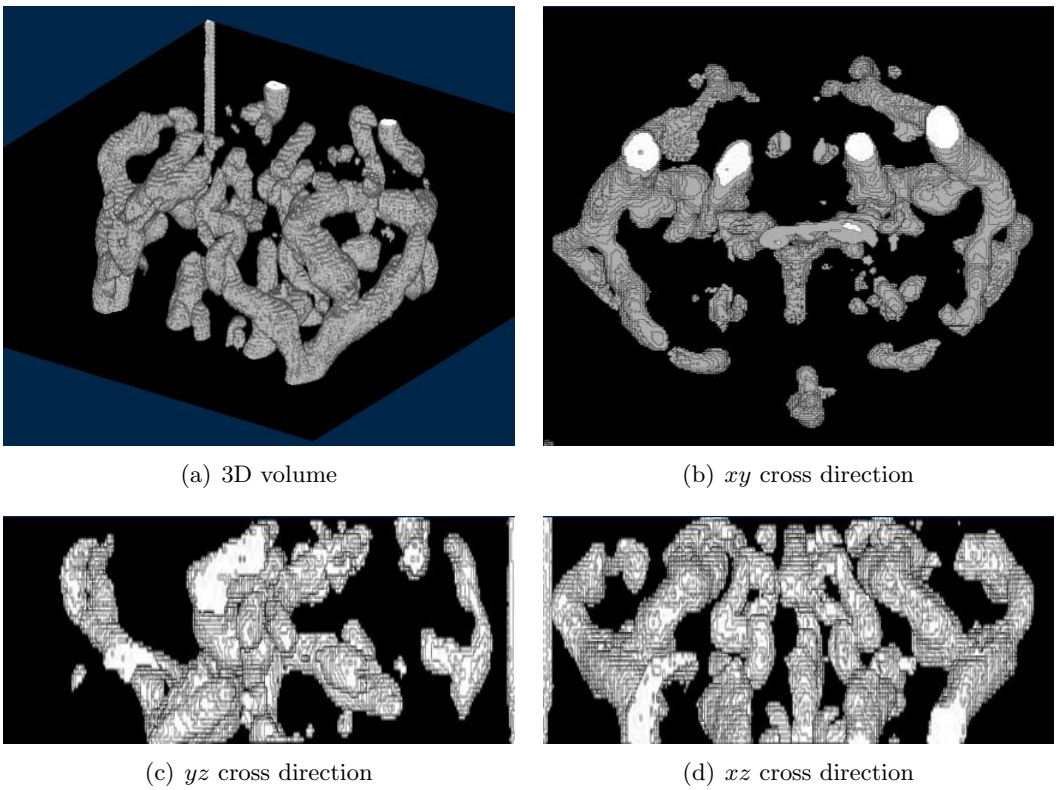
(c) yz cross direction

(d) xz cross direction

Figure 5: Detected contour shapes of 3D vessel volume.

The Figure 5 shows that the detected contour shapes give us a good reconstructed vessel structure and topology inside. It can be seen that the vessel's network is almost recovered though the structures are not connected enough. Vessels contrast is detected with a remarkable robustness, and most structures of vessels networks are preserved.

Depending on the success of our current strategy in two dimensions, we optimistic that we will be able to achieve closed contours in three dimensions. The closed contour detection is displayed in Figure 6, which is comparable to the detected contour shapes presented in 5.



(a) 3D volume (b) xy cross direction

(c) yz cross direction (d) xz cross direction

Figure 6: Closed contour shapes of 3D vessels volume based on Chamfer distance transformation.

Numerical experiments show that the proposed frame based model performs successfully closed segmentations of vessels. Three crossing directions xy, yz and xz in Figure 6 show closed contours that make the vessel network continuous. It can be seen that the discontinuous contour shapes in Figure 5 are well resumed.

5 Conclusion

In this paper, we have designed an algorithm of contour closing which based on the chamfer distance transform. The segmentation method is studied based on image distance transform and theory of saddle points, that requires a little of image calculation. Method gives the results that are good and enable the effective closure of the majority of gaps 5 to 10 pixels after the detection process.

Moreover, we conclude that such algorithm based on Chamfer distance transform is able to extend the contours along the gradient norm with candidate belonging to the image path. This method is well adapted to any types of two dimensional images, even if it includes textured or non-textured areas. Further testing on more demanding tasks such as vessel of mice brain segmentation are being conducted, especially in the case of three dimensional space. However, it could be noticed that our method may give some cases with false detection in discretization process, that need to be considered respect to the image histogram, that need to be approached automatically in the near future.

References

[1] Thiel E., Montanvert A., Discrete approximation of the Euclidean distance for image analysis: improvement of chamfer distances.

[2] Goldlucke B. (2010), Saddle Point Problems: Definition, properties and the $TV - \mathcal{L}^2$ model, Variational methods in Computer Vision II.

[3] Petrova J. and Hostalkova E. (2011) Edge detection in medical image using the Wavelet transform, Report of Research, Department of Computing and Control Engineering, Czech Public.

[4] Borgefors G. (1984) Distance transformations in Arbitrary Dimensions, Computer vision, Graphics and Image Processing, [27], 321–345.

[5] Milgram M., Cocquerez J.P. (1986) Fermeture des contours par un opérateur local, Traitement du signal [3](6), 303–311.

[6] Wirjadi O., Survey of 3D Image Segmentation Methods, Models and Algorithms in Image Processing.

[7] Minh-Phuong Tran (2012) 3D image analysis with variational methods and wavelets - Applications to medical image processing, PhD Thesis, École Doctorale Mathématiques, Informatique, Physique Théorique et Ingénierie des Systèmes, Laboratory MAPMO.