Amélioration des ouvertures par chemins pour l'analyse d'images à N dimensions et implémentations optimisées

François Cokelaer

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **SIGNAL, IMAGE, PAROLE, TELECOMS**

Arrêté ministériel : 7 août 2006

Présentée par

François COKELAER

Thèse dirigée par **Jocelyn CHANUSSOT**

préparée au sein du **Grenoble Images Paroles Signal Automatique Laboratoire (GIPSA-Lab)**
dans **l'Ecole Doctorale « Electronique, Electrotechnique, Automatique, Traitement du Signal »**

Améliorations des ouvertures par chemins pour l'analyse d'images à N dimensions et implémentations optimisées

Thèse soutenue publiquement le 22/02/2013
devant le jury composé de :

M. Fernand MEYER
Directeur de Recherche, CMM Fontainebleau, Président du jury
M. Christian GERMAIN
Professeur IMS Bordeaux, Rapporteur
M. Fabrice MERIAUDEAU
Professeur Université de Bourgogne, Rapporteur
Mme. Ingela NYSTRÖM
Professeure Université d'Uppsala, Examinateur
M. Hugues TALBOT
Professeur associé, ESIEE Paris, Examinateur
M. Paul GADER
Professeur Université de Floride, Examinateur
M. Mathieu FAUVEL
Professeur associé Université de Toulouse, Examinateur
Membres invités : M.Laurent BERNARD, Directeur Général Reactiv'IP et M. Olivier FRANCOIS, Directeur Technique CYXplus

Advanced Path Operators for the Analysis of n-Dimensional Images and Efficient Implementation

François Cokelaer
Supervisor: Jocelyn Chanussot
GIPSA-Lab - DIS
Université de Grenoble

Submitted in January 2013
PhD In Image Processing
Defended : February 22nd 2013

Abstract

The detection of thin and oriented features in an image leads to a large field of applications specifically in medical imaging, material science or remote sensing. Path openings and closings are efficient morphological operators that use flexible oriented paths as structuring elements. They are employed in a similar way to operators with rotated line segments as structuring elements, but are more effective as they can detect linear structures that are not necessarily locally perfectly straight. While their theory has always allowed paths in arbitrary dimensions, *de facto* implementations were only proposed in 2D. Recently, a new implementation was proposed enabling the computation of efficient d-dimensional path operators. However this implementation is limited in the sense that it is not robust to noise. Indeed, in practical applications, for path operators to be effective, structuring elements must be sufficiently long so that they correspond to the length of the desired features to be detected. Yet, path operators are increasingly sensitive to noise as their length parameter L increases. The first part of this work is dedicated to cope with this limitation. Thus, we will propose an efficient d-dimensional algorithm, the robust path operators, which use a larger family of flexible structuring elements. Given an arbitrary length parameter G, path propagation is allowed if disconnections between two pixels belonging to a path is less or equal to G and so, render it independent of L. This simple assumption leads to a constant memory bookeeping and results in a low complexity. The developed operators have been compared qualitatively and quantitatively to other efficient methods for the detection of line-like features. As an application, robust path openings have been integrated into a complete chain of image processing for the modeling and the characterization of glass fibers reinforced polymer. Our study has also led us to focus our interest on recent morphological connected filters based on

geodesic measurements. These filters are a good alternative to path operators as they are efficient at detecting the so-called "tortuous" shapes in an image which is precisely the main limitation of path operators. Combining the local robustness of the robust path operators with the ability of geodesic attribute-based filters to recover "tortuous" shapes have enabled us to propose another original algorithm, the selective and robust path operators.

Acknowledgements

It is with my greatest gratitude that I acknowledge the support and help of my supervisor Jocelyn Chanussot. He has trusted and encouraged me for three years and it was a pleasure to work with him and to learn from him.
I would like to thank the president of my oral defense committee, the reviewers and examinators. It was an honor to present my work in front of you.
I wish my thanks to Hugues Talbot, he has given me the opportunity to work on an interesting and quite open subject. Also thank you for helping me concerning the writing of articles.
I would like to thank my colleagues, especially Mauro Dalla Mura for its interesting talks and questions, it has really helped me to improve my research, Silvia Valero and Longyu Jiang who supported me a lot during three years.
Also, I would like to thank Lucie for her reliable support and for her impressive patience. I have found my greatest motivations thanks to you.

This research is supported by the DGCIS (General Directorate for Competitiveness, Industry and Services), the FUI (Interministerial Fund) within DELPIX project.

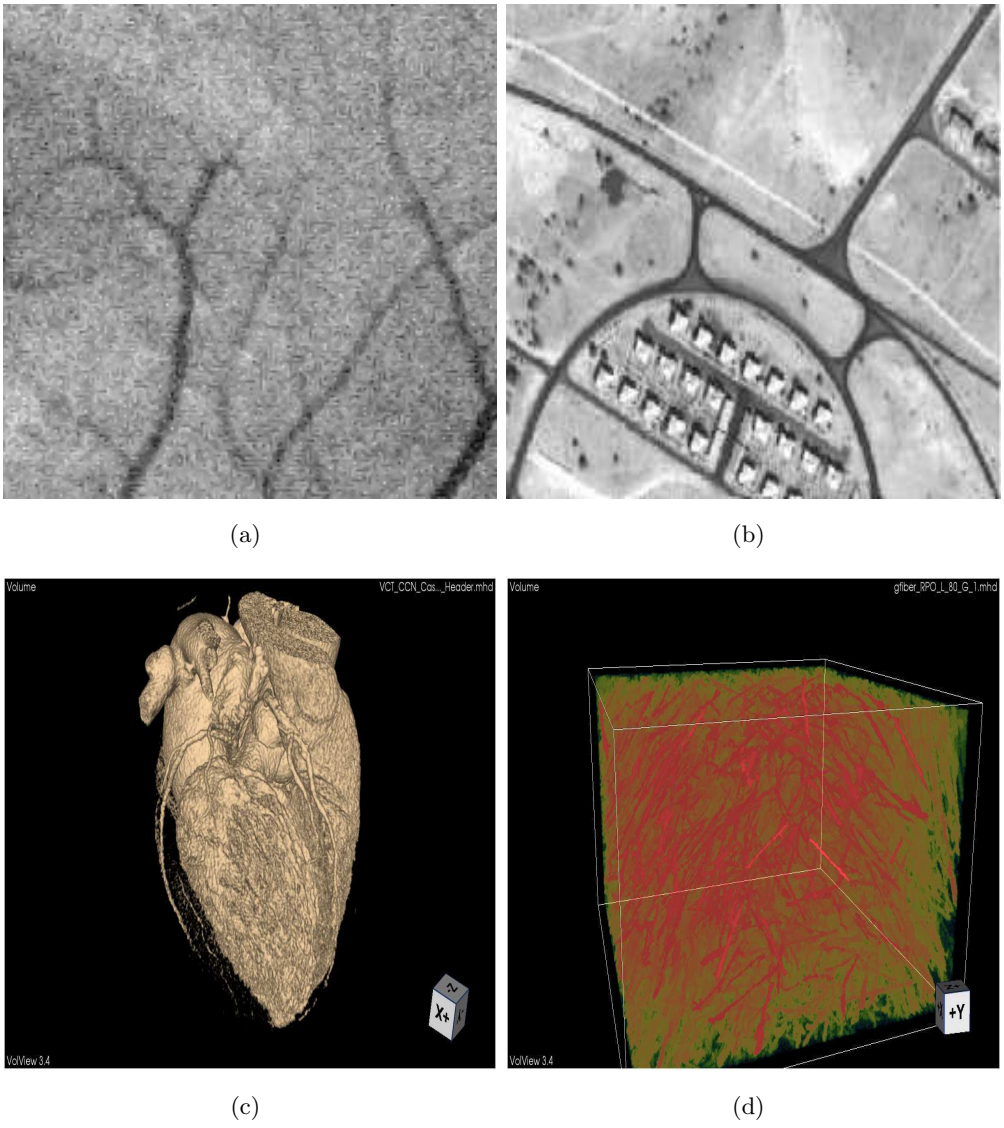
Contents

1

Introduction

With the emergence of new imaging systems we are now able to generate three dimensional (3D) images that enable us to see many phenomena occurring from the macroscopic to the nanoscopic scale.
Firstly designed for medical imaging, X-ray computed tomography (CT) has become widely spread as it performs a non-invasive mean for the practitioner to explore human body and to perform an accurate diagnosis of the diseases.
This imaging system has more recently found interesting applications in industry, especially in the domain of automotive and aeronautic where there is a need to predict the macroscopic behavior of the material by characterizing precisely its microscopic properties (e.g. orientations, lengths and shapes).
The drawback of this revolution relies now mostly on the question of "how can we automatically extract reliable information from the generated data ?"
In 2D, image processing and analysis have proven to be very efficient in many domains: medical and biomedical imaging, remote sensing, non-destructive testing, material science, biometric applications and so on.
Filtering in image processing is probably one of the more important task. Based or not on *a priori* information on image content, it aims at considering the useful information as "signal", and all that is not "signal" as "noise". Depending on the problem we are faced with, "signal" and "noise" can be of different nature.
Among all the methods provided by the scientific community for the last 40 years, the critical issue in image processing is to find what method best fits to our application.
At the cost of an increase in the computation time and the memory usage (mostly

because of the rise of the amount of data to process) 2D implementations of standard used image processing algorithms have been given 3D implementations. We can think about the well-known edge detector or smoothing methods for example.
However, to increase the reliability of the analysis in more and more complex cases of study, advanced methods should be used but often, these methods are involving a bulky implementation in 3D (e.g. non-local means algorithm).
To cope with this limitation, some algorithms have been developed with parallel implementations. These implementations improve considerably the computation time, however, it needs a dedicated machine to process the image. Moreover some interesting family of algorithms are not yet given a parallel version.
In this dissertation, we will propose several new algorithms for both 2D and 3D image processing based upon efficient ones for the detection of thin, curvilinear and oriented features. Especially, we will focus our interest in the case where thin features are noisy and are difficult to detect as they are affecting by disconnections (see Figure 1.1 for an illustration). After giving a review of the state of art of the detection of curvilinear features in image processing, our study will be focused on recent advanced filters provided by mathematical morphology called *path operators* (1). This algorithm is central in our work, indeed, we are dedicated to find a convenient way to increase the robustness of the so-called *complete* version to render the path openings and closings more efficient on real 2D and 3D applications.
We will introduce the robust path operators as a novel and efficient way to render the standard version of path operators robust to disconnections affecting the thin features in both 2D and 3D images. After giving the principles, the theoretical framework and an efficient d-dimensional algorithm to compute them, they will be used in the last part of this dissertation as a preprocessing step for the enhancement of a 3D image of glass fibers reinforced polymer.
Secondly, we will focus on the recently developed morphological connected filters based on geodesic measurements. These filters have also been proven to be efficient to detect thin and curvilinear features in an image. Even if they are for the moment not robust to noise, they have a real ability to detect the so-called "tortuous" shapes in an image. Fortunately it appears that the detection of "tortuous" shapes is a limitation of path operators algorithm, thus we propose to combine the two filters in an algorithm called the selective and robust path operators which aims at combining the local robustness



(a) (b)

(c) (d)

Figure 1.1: Example of applications: (a).Retinal vessel; (b).Roads in remote sensing; (c).Coronary arteries detection in medical imaging; (d).Glass fibers reinforced polymers in non-destructive testing

of path operators with the ability for geodesic-based filters to give a good response to tortuosity.

All the proposed algorithms proposed in this study have been implemented in C++ using the object oriented programming principle. The developed classes can be inter-

faced with any of standard image processing libraries providing basic read and write functions and a raw data extraction function. Here we have used the ITK library as it is convenient to read both 2D and 3D images of any type.
Particular cares have been given to propose efficient implementations runnable on a basic material configuration. Given timings in this document are achieved with a laptop computer with a dual-core processor of 2.2 GHz with 4 GB RAM memory and without any parallelization (e.g. when we will speak further in this dissertation about the computation of the path openings considering several orientations, these are computed sequentially).

2

Line-like Feature Detection

Detecting thin line-like features is still a real challenge in image processing. A huge number of methods from the past 40 years were dedicated to this task for both preprocessing and segmentation purposes as they play a key role in a lot of different applications (e.g., vessels detection in medical imaging is maybe one of the most famous ones). Here we are interested in the preprocessing methods aiming at increasing the useful information coming from thin and elongated features while removing the noise.

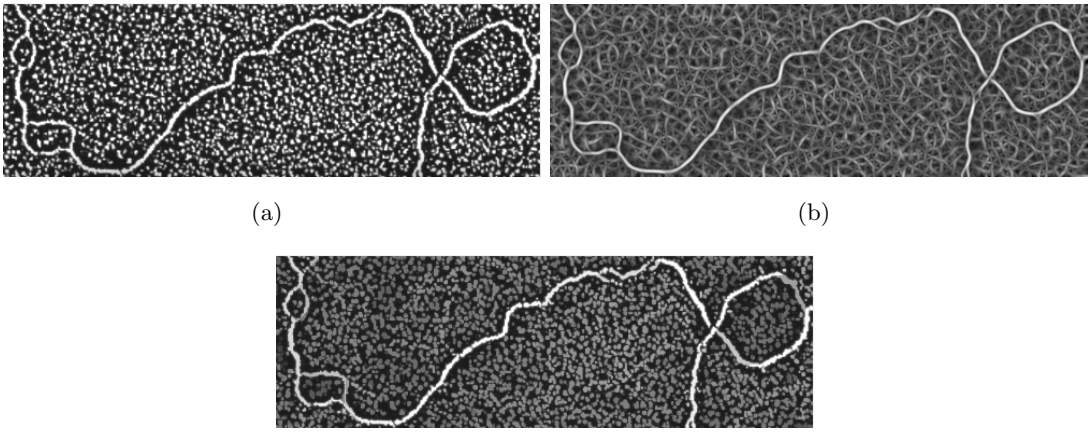
Even if their classification could be made in a lot of different manners, we have decided to make a separation between "preserving" and "detecting" filters.

"Preserving" filters will keep the features of interest intact while trying to remove most of the noise, whereas, the goal of the "detecting" filters is to expect a stronger response at features of interest and a weaker one at noise through a measure function (see Figure 2.1 for an example of a "preserving" and a "detecting" filter).

In this chapter we will recall the most widely spread methods for the detection of line-like features. More details are brought to the description of the methods provided by mathematical morphology as this discipline is central in our dissertation.

2.1 "Detecting" Filters

Most of the filters in this section are based on an intensity and shape model of line-like patterns. Indeed, it is assumed that the intensity of elongated features varies slowly



(a) (b)

(c)

Figure 2.1: Example of a "detecting" and a "preserving" filter (a).Original DNA filament; (b)."Detecting" filter (obtained with steerable filter); (c)."Preserving" filter (obtained with robust path operator);

along the direction of the ridge but presents higher contrast variations in its orthogonal direction.

2.1.1 Oriented Filters

Oriented filters are used in many of image processing applications: texture analysis, edge and ridge detection, image enhancement, features' orientations.
These operations are realized by performing a matched filtering considering a rotated version of an arbitrary chosen basis kernel. The finality is to examine filter's responses at many orientations (by rotating the basis kernel in several orientations) and to get the strongest one.
These filters are particularly suitable for the enhancement of vessels in medical imaging and has led to many of applications for images coming from various modality of acquisition (e.g. MRI (Magnetic Resonance Imagery), MRA (Magnetic Resonance Angiogram)).
Poli *et al.* (2) proposed a real-time implementation of these filters, sensitive to both ridges' orientation and thickness by combining shifted versions of Gaussian kernels. Obviously, the higher number of orientations are examined the more accurate will be the result of the filtering process and the higher will be the computational cost.

Based on the same idea, Kunz *et al.* (3) decided to use the result of the convolution of the image with highly anisotropic filters g_{θ_i} by rotating a basis kernel g in the orientation θ_i (the method needs a preliminary discretization step controlled by the number of rotated kernels one wants to apply on the image). Basically, a second order derivative of Gaussian is chosen as basis kernel to probe the contrast in and out the ridges. In addition to the width w of the ridges one wants to detect, an integration parameter l enables to reduce the noise by smoothing orthogonally along a discrete line for each of the rotated filters (see Figure 2.2).
These filters have been proven to be very efficient to detect low SNR (Signal to Noise Ratio) (4) and as the rotated kernel is separable, they are computationally very efficient. However, their adaptation to 3D image processing may not be straightforward particularly for the delicate issue coming from the discretization step.

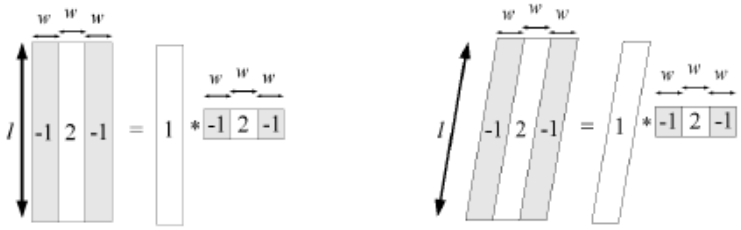


Figure 2.2: Left: The basis kernel; Right: A rotated version of the basis kernel

Steerable filters (5), (6) are a subclass of oriented filters in which the response at an arbitrary angle can be computed efficiently and analytically as a linear combination of a set of basis filters to cut down the computation time of a direct implementation approach (i.e. the implementation of each of the rotated version of the basis kernel as we have seen just above).
A function $h(x, y)$ is steerable if it can be written as a finite sum of weighted rotated version of itself:

$$h_\theta = \sum_{i=1}^{M} \omega_i(\theta) h^{\theta_i} \tag{2.1}$$

Derivatives of Gaussian kernel of all orders have been proven to be steerable (see Figure 2.3 for an example of a fourth-order Gaussian derivative) and are widely used as ridges detectors.

Jacob *et al.* (6) improves the previous work of Freeman *et al.* (5) by designing a family of oriented filters optimized according to a generalization of the Canny like criteria (7). Their filters provide the best compromise considering SNR, false detections and localization.

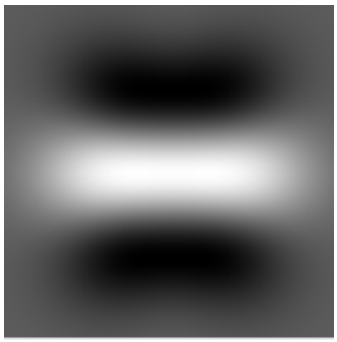


Figure 2.3: Ridge detector (fourth order derivative)

2.1.2 Hessian Based Filters

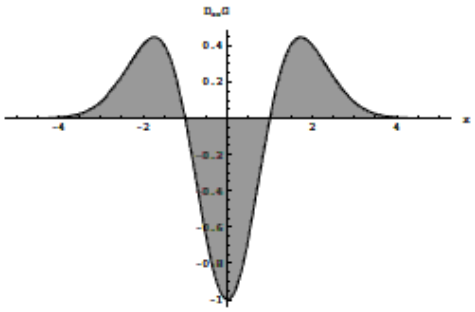


Figure 2.4: Second order Gaussian derivative

Considering a 2D image as an elevation map of $\mathbb{R}^2 \rightarrow \mathbb{R}$ derivatives methods are assuming that features of interest represent the crest lines of this elevation map (in the case of the detection of brighter features). Using kernels based on the widely used derivatives of Gaussian, one can access to the differential property of the image in several directions and then detect features presenting particular shapes (e.g. tubular objects for example).

The Hessian filter (8), (9), (10), (11), (12), (13), (14), is one of the widely spread filters in the community of medical imaging. The benefit of using the Hessian filter is mostly because of its property to be multi-scale. Indeed, performing the algorithm at different scales enables one to retrieve both smaller and larger features (note that this is very useful to cope with the randomness of anatomical structures).

Based on the linear scale-space theory (15), Hessian based filters are obtained by convolving the image with second order derivatives (see Figure 2.4) of Gaussian kernels of different standard deviations thus enabling the characterization of the local geometry of a pixel according to a given scale.

After building the Hessian matrix and extracting its eigenvalues (λ_i), a vesselness function can be used as a measure of likelihood for each of the pixel to be a part or not of a vessel (e.g. Frangi or Sato vesselness function). Figure 2.5 is an example of the result of the multi-scale Hessian filter based on the Frangi measure (14).

For a 3D image $I(p)$ with $p \in \mathbb{R}^3$, at a given scale σ, indicating the Hessian matrix eigenvalues as λ_1, λ_2 and λ_3 ($|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$) Frangi expressed its vesselness function as:

$$\nu(p,\sigma) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \text{ or } \lambda_3 > 0 \\ (1 - \mathrm{e}^{\frac{-R_A^2}{2\alpha^2}})\mathrm{e}^{\frac{-R_B^2}{2\beta^2}}(1 - \mathrm{e}^{\frac{-S^2}{2c^2}}) & \text{otherwise} \end{cases} \tag{2.2}$$

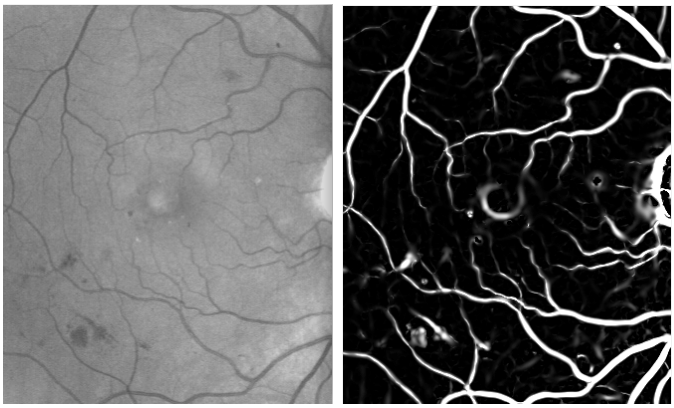
with $R_A = \frac{|\lambda_2|}{|\lambda_3|}$, $R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_1\lambda_3|}}$, $S = \sqrt{\sum_j \lambda_j^2}$ and α, β, γ parameters defined by the user. This vesselness function is computed in a given range of σ, σ_{range}, thus can give for each of the voxel a probability to belong to a ridge of a certain width.

After being normalized (in order to reach an equivalent response over the scales), one can select within σ_{range} the maximum of the response of the filter.

$$\nu_{max}(p) = \max_{\sigma_{range}} (\nu(p,\sigma)) \tag{2.3}$$

σ_{range} can be chosen so that it will cover a range of ridges widths that are expected to be found in the image.

The well known drawbacks of this method are its sensitivity to noise (due to the use of the second order derivatives) and its inability to detect junctions (causes disconnections in the features) as their characterization is very closed from that of blob like features. Moreover, at larger scales, the blurring effect of the Gaussian kernels can lead to geometric distortions which can be critical for some kind of applications (e.g. in vessels' medial axis position (16)).



(a) (b)

Figure 2.5: (a).Input image: 2D retinal image (b).Result of the multi-scale Hessian filter based on Frangi measure (14);

2.1.3 Curvelets

Curvelet transform is a recent multiscale directional method coming from the signal processing theory (17), (18), (19), (20), (21), which aims at overcome some shortcoming of wavelet transform by providing a sparse representation of curves.
Basically, for all the time-frequency methods used in signal processing it is possible to decompose a signal f in a linear combination of weighted basis functions:

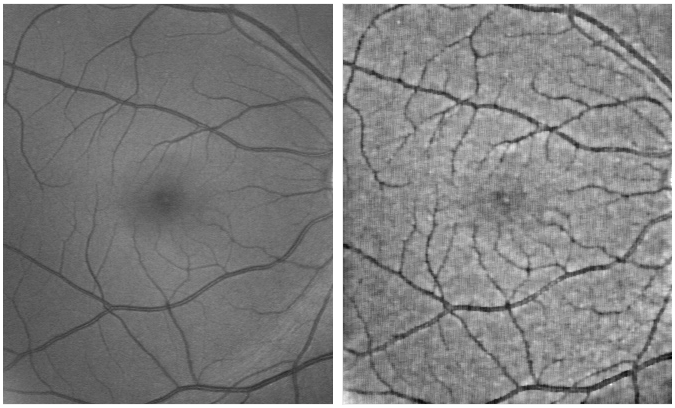
$$f = \sum_k \boldsymbol{a}_k b_k \tag{2.4}$$

with $\boldsymbol{a}_k$ and b_k the weighting coefficients and the basis functions respectively. This signal representation can be used to perform different tasks:

- data compression,
- feature extraction,
- image restoration.

There are two different implementations to compute the 2D discrete curvelet transform: the wrapping method and the unequispaced fast Fourier transform (21). The discrete curvelet transform is obtained by dividing the Fourier space into concentric circles and further dividing them into wedges. These radial wedges contain the structural and

anisotropic information of the image.
Since the curvelet transform is well suitable to give a sparse representation of the edges of an image, it can be used to enhance them by introducing a function which is in charge of modifying the curvelet coefficients and whose parameters are tuned according to image statistics (19). Figure 2.6 is an example of a non-linear enhancing of curvelet coefficient used in (22).



(a) (b)

Figure 2.6: Curvilinear features enhancement with curvelet transform

2.1.4 Minimal Path

Vincent (23), (24), proposed an efficient algorithm to enhance curvilinear features in a noisy environment. The principle of the local minimal path is described by the author as follows:
Assign to each pixel p of an image I the minimal cost $C_I(p)$ for all the paths P of a given length l originating from p and whose orientation and straightness are within a given range.
Recently, Bismuth *et al.* (25), generalized the concept of the local minimum path of Vincent (24) by adding a constraint on its smoothness. They are making the parallel between the classical model based filters (e.g. matched filters) and the proposed polygonal path which aims at selecting for each pixel a best-fitting curve of given smoothness and length.

This path, defined by two parameters (the arclength L and the smoothness l) is in fact, for $l = 1$, a particular case of the local minimal path of Vincent. Moreover, as shortest paths from random positions tend to form bundles around the curvilinear features of interest, they also made use of a voting scheme ((26),(27), (28)) to reinforce path overlapping. This family of algorithms have found its applications in medical imaging especially in guide-wire detection where the SNR is very low or in the extraction of the medial axis of coronary arteries in 3D (16).

2.2 "Preserving" Filters

2.2.1 Isotropic Filters

Isotropic filters are based on the assumption that image values vary slowly over their spatial domain. Thus, it seems logical, as all the pixels in a given neighborhood should be similar, to average them together. Making the hypothesis that the noise affecting the pixels in a neighborhood is less correlated than the signal, the averaging process should decrease the amount of noise while keeping intact the useful signal. As an output, the well-known Gaussian filter is performing a weighted sum of the neighborhood pixels values depending on the distance from the central pixel of the neighborhood.
Even if this idea is the foundation of filtering in image processing, it is very limited as the hypothesis of slow spatial variations fails at image edges. This effect increases in the case of the filtering of thin and noisy features where classical isotropic kernels will not fit into these particular kind of shapes as it is hard to find "low gradient" regions in this case.

2.2.2 Edge-preserving Filters

2.2.2.1 Bilateral Filter

As seen above, classical fixed-shape kernels blur the edges of the image and are so particularly not well adapted to the enhancement of thin and noisy curvilinear features in a image. Edge-preserving filters aim at smoothing homogeneous regions while trying to keep intact the information provided by the edges (e.g. (29), (30) are using a statistic homogeneity measure to combine similar pixels together within a given neighborhood). Here we will give some details about one of the most well-known edge-preserving filter:

the bilateral filter. Based on the early work of (31), the bilateral filter, defined in (32), is an edge preserving filter which combines both photometric (or range) and domain filtering.

The filtering power will depend on two parameters σ_r and σ_d in the particular case of shift-invariant Gaussian kernel. The principle of this filter is very intuitive and so are its parameters to set. Given a pixel p, its value will be replaced by the weighted values of the pixels similar to p in intensity and in the spatial vicinity of p.

The weight given to a pixel belonging to the neighborhood of p will depend on the geometric spread σ_d and the range spread σ_r. A larger σ_d will combine values from more distant pixels' locations and would blur more the image. A larger σ_r will enlarge the set of combination by considering pixels with larger intensities' differences from p. The intensity I_p of pixel p will be replace by :

$$I_p^{bf} = \frac{1}{W_p^{bf}} \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q \tag{2.5}$$

with G_{σ_d}, G_{σ_r} a Gaussian convolution kernel of variance σ_d and σ_r respectively and q a pixel in the vicinity of p. W_p^{bf} normalizes the sum of range and domain weights:

$$W_p^{bf} = \sum_{q \in S} G_{\sigma_d}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) \tag{2.6}$$

In the case of a large smooth region, the bilateral filter will act like a classical domain filter (Gaussian filter of parameter σ_d); it averages the small intensities differences between pixels caused by noise as they are weakly correlated compare to “signal” pixels. In the case of a sharp edge between two regions (e.g. step between a dark and a bright region), considering a pixel p on the bright side of the boundary, the similarity criterion defined by σ_r will prevent combining the pixels which are located on the dark side of the boundary but will suppress the noise on the bright side. Here we will use the efficient implementation proposed by (33) based on a signal processing approach.

2.2.2.2 Nonlinear Anisotropic Filtering

Within the framework of the scale space theory, given an image I, a family of images can be generated from a finer to a coarser scale according to the well-known diffusion equation:

$$I_t = \nabla.(D \nabla I) \tag{2.7}$$

With D the diffusion tensor enabling to control the blurring process to preserve or not features of interest and $I_0 = I$. If D is the identity matrix, such equation becomes the classical "heat-equation" which is satisfied by the particular case of the Gaussian kernel. However this filter does not preserve the contours and it smooths the image in an anisotropic way regardless of the presence of the contours.

In order to increase the intra-region smoothing while preserving the edges in the image, Perona and Malik introduced the non-linear anisotropic diffusion (34) in which D is replaced by a decreasing function of the local gradient magnitude.

Based on this idea, many of authors contributed to anisotropic diffusion filters by proposing more elaborated diffusion tensor matrix (35), (36), (37). For example Catté *et al.* (38) proposed to consider the norm of the gradient of the smoothed image (image blurred by a Gaussian kernel) instead of the norm of the gradient, Krissian *et al.* (39), (40) suggest to reinforce the diffusion process of the image in the direction of the lowest curvature.

In another interesting fashion, Orkisz *et al.* (41) decline the non-linear anisotropic filtering using some kind of oriented structuring elements called "sticks" in which a smoothing is performed (note that these "sticks" have their equivalent in mathematical morphology with oriented segments).

The local orientation is estimated thanks to the intensity homogeneity along the "sticks" and to the difference of intensity between each of them. Obviously, this step is preceded by a discretization on the orientation of the "sticks".

The same approach was also used by several authors in (42), (43), (44), (45), (46) and (47). They were used first as an efficient mean to enhance the vizualisation of MRA images by maximum-intensity-projection (MIP) where the thin vessels are not well contrasted.

This family of filters enables the smoothing to be really efficient along the vessels, nevertheless it shows some limitations for the enhancement of the strongly curvilinear ones. Moreover, as these filters are not multi-scale they don't have the capability to adapt to both smaller and larger scales. Based on the same idea, Truc *et al.* (48) proposed to cope with these limitations with a *decomposition-filtering-recombination* method. The decomposition of the image into several "directional images" provided by a directional filter bank is following by an enhancement step (using a Hessian based

filter) and then by a recombination step enabling the reduction of the sensitivity to noise and to a better detection of the junctions patterns.

2.2.3 Mathematical Morphology

2.2.3.1 Basic Concepts

Mathematical morphology (49), (50), (51), (52), (53), (54) relies on the *set theory* that make it “self-encompassed” and coherent. This theory is well adapted to describe the world around us since the visual information is not translucent and is *a contrario* composed of opaque objects that hide one another.
The main issue concerning mathematical morphology is to extract information from a non-linear transform applied to objects in image. This information can be, but are not limited to:

- shape,
- orientation,
- size,
- connectivity ...

L. Najman and H. Talbot (55) well summarize this discipline: *Mathematical morphology was historically the first non-linear theory in the field of image processing. It rests on three pillars that make its success: a solid theory, a wide scope, and an effective implementation.*
Here we will give the basic theoretical concepts of mathematical morphology. The reader that is already familiar with these concepts can skip this section.
In order to define operators for mathematical morphology, we make use of the abstract notion of *complete lattice*.
A *lattice* $(E, \leq)$ is defined as a set E equipped with an ordering relation $\leq$ reflexive ($x \leq x$), antisymmetric ($x \leq y$ and $y \leq x \Rightarrow x = y$) and transitive ($x \leq y$ and $y \leq z \Rightarrow x \leq z$). This relation enables the definition of a “greatest” element ($x \vee y$) and a “lowest” element ($x \wedge y$). A *lattice* is said to be *complete* if each of the subset P of E admits a smallest majorant $\vee P$ (*supremum*) and a largest minorant $\wedge P$ (*infimum*). Basis operators are derived from the notion of complete lattice and from the fact that

they respect its fondamental structure, i.e. preservation of the order ($x \leq y \Rightarrow \phi(x) \leq \phi(y)$) and commutativity with the supremum and the infimum. Specifically, the dilation δ is defined as commuting with the supremum and the erosion ϵ as commuting with the infimum.
We can give some classical examples of complete lattice:

- $\mathbb{R}$, indeed an arbitrary interval admits a *supremum*, and an *infimum*. Moreover the ordering relation $\leq$ is a total ordering relation (two arbitrary chosen elements of $\mathbb{R}$ can always be ordered).
- The set $\mathcal{P}(E)$ of all the subsets of E which is ordered by the inclusion operator and in which *supremum* and *infimum* coincide with the union and the intersection.
- Functions having their values in $\mathbb{R}$ or $\mathbb{Z}$ in which the *supremum* and the *infimum* is defined as the pointwise maximum and minimum respectively.

In a complete lattice, *supremum* and *infimum* play symmetric roles. This duality is the main issue of mathematical morphology as it encompasses all its theoretical aspects. If we place ourselves in the lattice $\mathcal{P}(E)$, two operators ϕ and ϕ^* are said to be dual if for all X, $\phi(X^c) = [\phi^*(X)]^c$ where X^c is the complementary of X in E.
Fundamentals property of the basis operators $\phi : E \to E$ are:

- increasingness $x \leq y \Rightarrow \phi(x) \leq \phi(y)$,
- extensivity or anti-extensivity ($x \leq \phi(x)$ and $\phi(x) \leq x$ respectively),
- idempotence ($\phi(\phi(x)) = \phi(x)$).

In image processing, we would like to model the lattice of the binary image defined as mapping of: $\mathbb{Z}^2 \to \{0, 1\}$. It can be considered as a subset of a grey-level image E obtained by a thresholding operation. To make it easier to define the different operators, it is often convenient to place ourselves in the boolean lattice $\mathcal{P}(E)$, where E is $\mathbb{R}^n$ or $\mathbb{Z}^n$ equipped with a translation and where the *supremum* coincide with union and the *infimum* with the intersection. A basis operator is an operator preserving the laws of the lattice, specifically, in practical applications, image processing often involves the use of set B called *structuring element* whose shape is adapted to a given purpose (e.g. basic morphological filtering uses square or disc shapes).

Considering X a binary image, subset of E, the *shifted* set of X by $p \in E$ is defined as $X_p = \{x + p | x \in X\}$. The dilation of X by a set B is then defined as:

$$\delta_B(X) = X \oplus B = \bigcup_{b \in B} X_b = \{x + b | x \in X \text{ and } b \in B\} \tag{2.8}$$

Dilation is the geometrical *locus* of points z in E such that B_z intersects X. Basically, the dilation operation is thickening the set X with B. It respects the property of being increasing, extensive and commutable by the union operator.
Erosion is dual by complementation to erosion and is defined as:

$$\epsilon_B(X) = X \ominus B = \bigcap_{b \in B} X_{-b} = \{p \in E | B_p \subseteq X\} \tag{2.9}$$

It represents the geometrical *locus* p of E such that B_p is completely included in X. Basically, the erosion operation is thinning the set X with B. It respects the property of being increasing, anti-extensive and commutable by the intersection operator.
Performing a dilation on a set X with a structuring element B is equivalent to performing an erosion on X^c the complementary of X with the reverse structuring element.
Even if “filtering” in image processing is a generic term including all the possible treatments, filtering in the sense of mathematical morphology is involving a specific class of operators respecting precise properties. Indeed filtering operator must respect the property of being increasing and idempotent meaning that it converges into one iteration ($\phi(\phi(X)) = \phi(X)$).
Morphological openings (resp. closings) are morphological filters and are defined by composition of an erosion followed by a dilation (resp. dilation followed by an erosion). In the opening case, the dilation (computed with the reflected structuring element) is trying to reconstruct the features that were not totally suppressed by the erosion. Thus, the opening and the closing are defined as:

$$\gamma_B(X) = X \circ B = (X \ominus B) \oplus B \tag{2.10}$$

and

$$\phi_B(X) = X \bullet B = (X \oplus B) \ominus B \tag{2.11}$$

Note that openings (resp. closings) that can be defined by composition of erosion followed by dilation (resp. dilation followed by erosion) are termed as morphological

openings (also called *structural*). If a transformation have the same properties of an opening but can not be defined as well, it will be termed *algebraic* opening. Openings (resp. closings) are increasing, idempotent and anti-extensive (i.e. $\gamma_B(X) \subseteq X$) (resp. extensive i.e. $X \subseteq \phi_B(X)$).

2.2.3.2 Structuring Elements Based Filters

In the particular case of the filtering of thin elements, basic anisotropic shapes (e.g. disc, square ...) will remove too much of the features of interest. Thus, it is needed to adapt the structuring element according to the content of the image.
The powerfulness of basis morphological operators i.e. erosion and dilation is that they can be combined to build other more powerful operators. This is for example the case of morphological openings and closings.
Another example is that of operators taking care of elements in the foreground and in the background at the same time. Thus, we need two different structuring elements $T1$ and $T2$ applicable to foreground and background respectively. This transform applied on a set X is noted as:

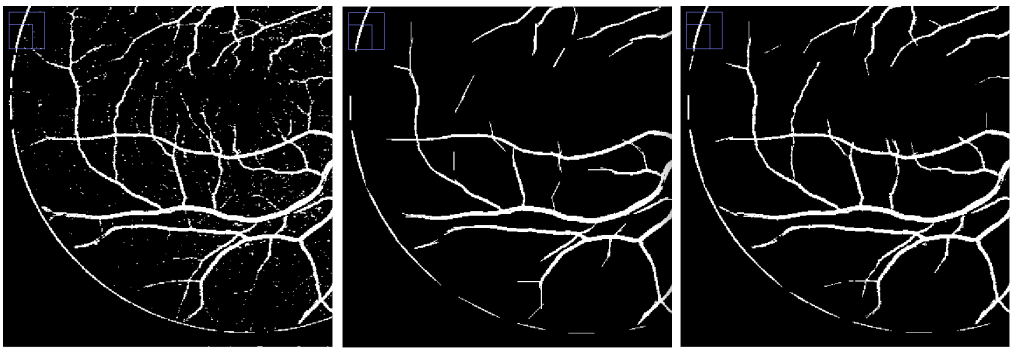
$$X \star T = (X \ominus T1) \cap (X^c \ominus T2) \tag{2.12}$$

Theses transforms are termed as *hit or miss transform* (HMT), (52), (56), (57), (58), (59), and are well performing the difficult task of shape recognition. Indeed, $X \star T$ is the *locus* p of E where $T1$ and $T2$ fits to the foreground and the background respectively. This transform is not increasing and so does not constitute a morphological filter. However, it has shown in its grey-scale extension a real ability to be efficient in practice e.g. for the extraction of coronary arteries in 3D medical imaging (57).
When performing an opening (resp. closing) by a given structuring elements, the brighter (resp. darker) elements that do not fit into the arrangement of the structuring elements will be removed.
Performing the union of the response of oriented segments within a large range of orientations should be a good answer to the problem of the detection of thin features especially when the orientations in the image are unknown.
Soille *et al.* (60) proposed an efficient algorithm to compute erosions and dilations along Bresenham lines (61). Latter, Soille and Talbot improved this algorithm rendering it invariant to translation (62) with quasi no additional complexity (applications can be

found e.g. in (63) and (64) for the detection of roads in remote sensing images and in (65) for the computation of the local structure orientation in 2D).
A “rank-based” (66) version of this algorithm was also proposed enabling the computation of openings with incomplete oriented line segment (i.e. with a given number of points lying into the background of the image): given a structuring element B whose cardinality card(B) equals n points, the rank-max opening of parameter r is equivalent to the union of all the structuring elements B_i included into B and containing r pixels:

$$\gamma_{B,r} = \bigvee_i \{\gamma_{B_i} \mid B_i \subseteq B \cap \text{card}(B_i) = r\} \tag{2.13}$$

Compared to the use of plain line segments, rank-max openings are less sensitive to the presence of small gaps (indeed gaps up to $(n - r)$ are allowed along the considered oriented line segment). This reinforces their applicability to the detection of features in noisy environment (see Figure 2.7). More recently, path openings and closings were



(a) (b) (c)

Figure 2.7: Comparisons between openings by line segments and rank-max openings, parameters are set manually depending on the considered application (a).Input image; (b).Result of the union of line segments ($L = 20$ and $\Delta_\theta = 15$); (c).Result of the rank-max openings using line segments ($L = 20$, $\Delta_\theta = 15$ and $r = 0.8$), disconnected straight parts of the vessels are detected

introduced in order to generate a family of oriented and flexible structuring elements. As this topic is the main topic of our dissertation, more details will be brought in the next chapter. Note that the work of Soille and Talbot (62) is important for our study as it is the starting point of our reflexion on the addition of robustness to path operators.

2.2.3.3 Mixed Approaches

The theoretical framework of spatially variant mathematical morphology (67), (68) allows to generate morphological operators whose shape and orientation can be locally adapted to the underlying directions of the image. In the work of Tankyevych *et al.* (69), (70), (71), (72), the curvilinear morpho-hessian filter computes the orientation information from the Hessian matrix. This filter has been used to enhance and reconnect vessels in 3D medical imaging. Close to the results obtained with the morpho-hessian filter, (73), (74) proposed an anisotropic filter where oriented structuring elements vary over the space according to a vector field. Note that this approach can be related to the early work on anisotropic non-linear filtering made by Orkiz in (41).

2.2.3.4 Connected Filters

Wilkinson *et al.* proposed shape preserving connected filter (the filtering process is realized by computing an attribute on each of the connected component of the image) (75) based on the volume and on the moment of inertia of the connected components in order to retain filamentous objects while removing the noise.
Recently Morard *et al.* (76), (77) generalized attribute thinnings (openings without the increasingness property) to geodesic measures performed on the connected component. From the geodesic diameter, they have derived several attributes e.g. length, elongation, tortuosity of a connected component.
Here we have just briefly mentioned these methods as more detailed explanations about connected filters will be provided in the chapter 5.

2.3 Discussion

In the case of the filtering thin and elongated features in a noisy environment, the preprocessing step aiming at retrieving the relevant information while reducing the noise is critical to perform in further steps the segmentation and the analysis.
As we have seen above, all of the methods rely on a model of what one wants to detect in an image.
Based on the scale-space theory, the "detecting" methods have proven their efficiency, however, many of these methods have limitations, either generating distortions of small details or enhancing the noise (i.e. non-elongated features).

Specifically, these methods are almost all relying on the use of a Gaussian kernels (or its derivatives) in order to have access to different level of details of the image. This can be critical for the efficiency and the semantically correctness of the result of the segmentation (e.g. position of edges and ridges).
Mathematical morphology provides a large choice of "preserving" filters which aim at conserving the radiometric and the geometric properties of the features of interest while trying to remove the noise. Moreover, a real effort was brought by the authors in the last decade at both algorithmic and implementation levels to reach satisfactory computation times and memory use in order to process larger and larger 3D data. This is what motivated our work on advanced morphological filters.

3

Path Operators

3.1 Motivation

In many situations, image processing involved the use of structuring elements based filters. These structuring elements act like a probe to compare their shape to the shape of the objects present in the scene.

For these operators to be efficient, structuring elements should be adapted to what one want to keep or not in the image after filtering. However, in practical applications, a very few kind of shapes are used. Anisotropic structuring elements (e.g. square or diamond shape) of different sizes are widely used for basic filtering, or granulometries (52), (78), (79).

When dealing with oriented and line-like features, one would use a union of discrete oriented line segments as structuring elements to filter out compact noise while not removing elongated features. However, the use of discrete lines segments is very limited to some particular applications as most features in real-world images are locally not perfectly straights and would not fit into a straight structuring element even if a large number of oriented segments are used. Moreover, line-like filters can be very time consuming (80) and, as we are interested in applications for 3D images, can be prohibitive as the number of orientations depends quadratically on the segment length L (81).

To cope with this limitation, path openings and closings were first proposed by Buckley and Talbot in (82) by adapting the efficient algorithm developed by Vincent (24). However the "shortest paths" algorithm of Vincent is based on local sums of pixels values and does not constitute a morphological filter.

In comparison, we will see below that path openings respect the property of being an algebraic opening and enable the generation of a family of structuring elements which allows oriented, narrow and elongated features to be flexible locally.
Later, path operators were given a more detailed theoretical framework and a recursive implementation in $O(NL)$ (with N and L image dimension and path length respectively) (1) by the same authors. An efficient implementation in $O(N \log(L))$ was proposed by Talbot and Appleton (83) in an ordered fashion. This ordered implementation constitute the basis of the work that will follow on robust path openings and closings. Recently, Cris L. Luengo (81) developed a dimensionality-independent version of Talbot and Appleton's implementation by simplifying the algorithm thus render it easier to implement. More details will be brought to this implementation as it is central in our study.

3.2 Principle

Given two parameters, L the length of the path in pixels and a local adjacency relation setting the global orientation of the path, path openings (resp. closings) principle is to remove features that are brighter (resp. darker) than their immediate surrounding and don't fit into an arrangement of pixels defined by the generated path. Performing a path openings is equivalent to the union of the openings computed from the family of all the possibilities of path arrangement.
With no *a priori* on the orientation on the content of the image, path openings are basically computed sequentially considering several global orientations. Unioning the results obtained from the different orientations performs a global anisotropic filtering.

3.3 Definition

Detailed theoretical foundations of path openings and closings can be found in (1). Here we will give a summary of the theoretical aspects for the reader to be able to understand the concepts behind these operators.

3.3.1 Adjacency Graph And Dilation

Starting from a set of points E (i.e. pixel locations in image domain), we define a spatial adjacency graph (e.g. Figure 3.1) between these points from a binary adjacency relation $x \to y$. Note that this binary relation is neither reflexive nor symmetric (symmetric would mean $x \to y$ iff $y \to x$, for every x, $y \in E$ and reflexive would mean $x \to x$ for every $x \in E$).
The set of all these relations specify the edges going from x to y. In this context, y is called the *successor* and x the *predecessor*.

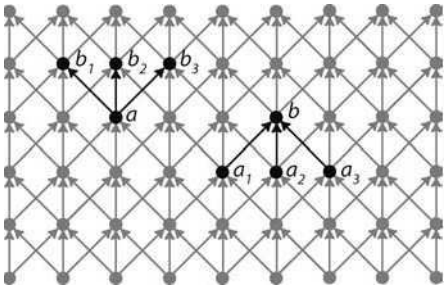


Figure 3.1: Illustration of a spatial connectivity graph defined by a binary adjacency relation; $b1$,$b2$,$b3$ are the successors of a and $a1$,$a2$,$a3$ are the predecessors of b.

Using this adjacency graph, we define for each point x the set of its *successors* with respect to $\to$ by :

$$\delta(\{x\}) = \{y \in E : x \to y\} \tag{3.1}$$

We can generalize this expression to an arbitrary subset of E by:

$$\delta(X) = \{y \in E : x \to y \text{ for some } x \in X\} \tag{3.2}$$

Analogously we can define the reverse operator:

$$\breve{\delta}(X) = \{y \in E : y \to x \text{ for some } x \in X\} \tag{3.3}$$

$\delta(X)$ and $\breve{\delta}(X)$ include all points that have a *predecessor* and a *successor* in X respectively. Note that δ and $\breve{\delta}$ have the algebraic property of a dilation i.e. preservation of the order and distribution over the *supremum* ($\delta \cup_i X_i = \cup_i (\delta X_i)$). One can notice that

it is relevant to place ourselves in in the complete lattice $\mathcal{P}(E)$ and that we can express the adjacency relation on this set as:

$$x \to y \text{ if } y \in \delta(\{x\}) \tag{3.4}$$

3.3.2 The Binary Path Operator

From the adjacency graph defined just above we can define a δ-path of length L which is a L-tuple:

$$\boldsymbol{a} = (a_1, a_2, ..., a_L) \text{ if } a_k \to a_{k+1} \tag{3.5}$$

Which is equivalent to:

$$a_{k+1} \in \delta(\{a_k\}) \text{ for } k = 1, 2, ..., L-1 \tag{3.6}$$

If $\boldsymbol{a}$ is a δ-path of length L, we define the corresponding reverse path:

$$\check{\boldsymbol{a}} = (a_L, a_{L-1}, ..., a_1) \tag{3.7}$$

called $\check{\delta}$-path of length L.
We can deduce that $\boldsymbol{a} = (a_1, a_2, ..., a_L)$ if and only if $\check{\boldsymbol{a}} = (a_L, a_{L-1}, ..., a_1)$ and both of them have the same length L. Considering the space of image domain E, we define the set of δ-path of length L in E by Π_L and conversely $\check{\Pi}_L$ as the set of all $\check{\delta}$-path of length L in E. Then from a path $\boldsymbol{a}$, we define $\sigma(\boldsymbol{a})$ the set corresponding to path elements: $\sigma(\boldsymbol{a}) = \sigma(a_1, a_2, ..., a_L) = \{a_1, a_2, ..., a_L\}$. We denote the set of all δ-path in a subset X of E by:

$$\Pi_L(X) = \{\boldsymbol{a} \in \Pi_L : \sigma(\boldsymbol{a}) \subseteq X\} \tag{3.8}$$

and conversely the set of all $\check{\delta}$-path of length L in X by $\check{\Pi}_L(X)$.
Path openings are defined as the union of all δ-path of length L contained in X:

$$\alpha_L(X) = \bigvee \{\sigma(\boldsymbol{a}) : \boldsymbol{a} \in \Pi_L(X)\} \tag{3.9}$$

$\alpha_L(X)$ satisfies all the algebraic properties of an opening: increasingness, anti-extensivity, idempotence.
In the case of a periodic adjacency graph (Figure 3.1 is an illustration of a 3-neighbors vertical periodic adjacency graph), $\alpha_L(X)$ is equivalent to the *supremum* of openings

over the generated family of structuring elements.
Moreover the use of a translation invariant graph will lead to the generation of a family of translation invariant structuring elements usually used for the practitioner in mathematical morphology (see Figure 3.2) .
Note that we have explained the theoretical aspects of path openings. Thus path closings are defined similarly by complementation, as usual in mathematical morphology. In practice, in 2D, a commonly used graph is a periodic 3-neighbor adjacency relation

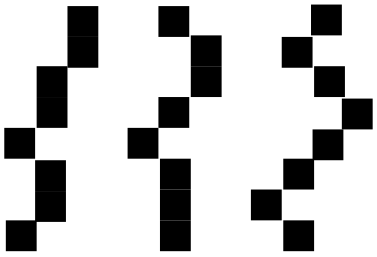


Figure 3.2: Example of three structuring elements of length 8 pixels generated by a periodic 3-neighbors adjacency graph

(see Figure 3.3). Basically, four path openings with four different orientations (one vertical, one horizontal and two diagonals) are computed sequentially. Unioning all these results enables the anisotropic detection of features of interest which is particularly useful when no *a priori* is given on the orientation present in the image. For a

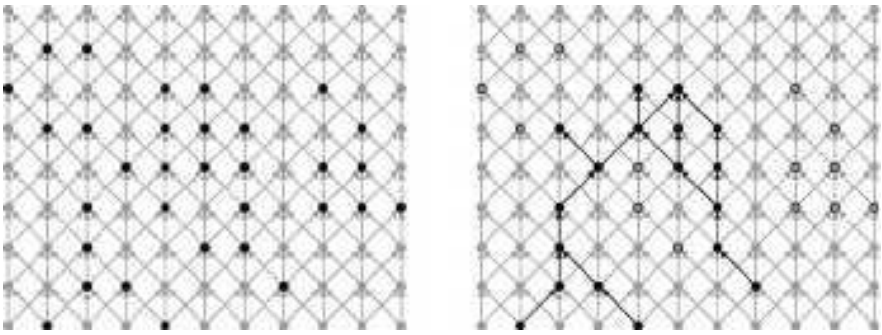


Figure 3.3: A set $X \in E$ representing the "active" elements set (left) and its openings $\alpha_6(X)$ (right). Adjacency graph is realized using a periodic 3-neighbor North-South adjacency relation. Elements left unfilled are discarded by the opening.

given length L, the number of structuring elements generated grows exponentially with

L (3^{L-1}). Thus, performing a morphological path openings is equivalent to union the result of the openings performed considering the 3^{L-1} structuring elements.
A direct approach which would consist in performing the union of openings computed with each of the possible arrangements would be completely prohibitive (e.g. for $L = 10$, the number of structuring elements is 19683).
However, we will see further in this dissertation that original algorithms have been designed to enable the efficient computation of path openings.

3.4 Extension To The Grey-scale Case

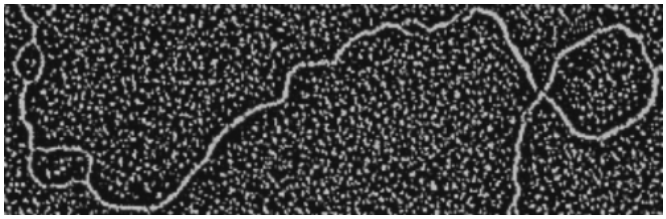
The extension of path openings to the grey-scale case is developed using the principle of threshold superposition (52), (84). The grey-scale path openings of a grey-scale image I is defined as:

$$\Pi_L^t(I) = \Pi_L(X_t(I)) \tag{3.10}$$

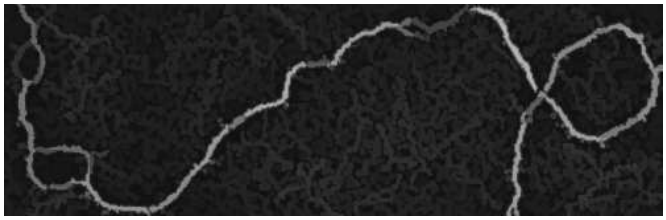
where $X_t(I) = \{x \in E : I(X) \geq t\}$ is the *level set* of I at a value t.
Roughly speaking, the grey-scale path openings (resp. closings) give for each of the pixels the highest (resp. lowest) value for which the binary path openings (resp. closings) are *true*. A grey-scale image can be then seen as a stack of binary sets, each of them represents the set of pixels present at a given grey-level. Applying the binary path openings on each of the binary set and then stacking them enables to reconstruct the result grey-scale image (see Figure 3.4).

3.5 Implementation Issue

Latest works on path openings were dedicated to find a mean to render its implementation tractable for 2D and then 3D image processing applications.
The first implementation able to compute a 2D complete and incomplete path openings for both binary and grey-scale images was given in (1). Based on a recursive implementation the complexity involved was in $O(LN)$ thus was penalized by longer path lengths. In (83), Talbot and Appleton have proposed an efficient implementation again for both complete and incomplete path openings in an ordered fashion thus have succeeded to reduce the complexity to $O(Nlog(L))$ leading to faster implementation



(a)



(b)

Figure 3.4: (a).Input grey-scale image; (b).Result of path openings with $L = 60$;

than classical algorithm performing the *supremum* over a family of oriented segments. Here we will give the details (see Figure 3.5) of the dimensionality-independent version of Talbot's algorithm proposed by Cris L. Luengo Hendriks (81) because it gives a flexible implementation framework that will be used latter in this study. Dimensionality-independent path operators (81) are based on the principles behind complete ordered path operators given by Talbot and Appleton in (83) but provide a more convenient implementation suitable to process n-d images.

One could correctly implement a grey-scale path openings (resp. closings) by stacking the results of all binary path operators from the lowest to the highest threshold (resp. highest to lowest). At the end of the process, the value of the stacked operator at a particular point corresponds to the highest threshold for which the binary operator remains true at that point (meaning that at this grey level, this pixel fits into the structuring element defined by the length and by the adjacency relation).

Here we provide some implementation details about this algorithm. For an opening (resp. a closing) the algorithm starts by creating a linear array of pixels memory addresses ordered by grey level value (low to high for opening and high to low for closing). Starting from the lowest to the highest grey level value, each of the pixels that is still active will be treated independently and will be considered as a "seed" to

```
create list of offsets n+ to upstream neighbors and n− to downstream neighbors
create a list i of indices to every pixel in the image I (except border pixels)
sort i according to value of I(i)
create temporary images b, λ+ and λ−
initialize: b ← true, λ+ ← L and λ− ← L
initialize: b(p_b) ← false (for all border pixels p_b)
for every element p in i for which b(p) = true
    propagate(p, λ−, n+, n−)
    propagate(p, λ+, n−, n+)
            for every element q in Q_c
                if λ+(q) + λ−(q) − 1 < L
                    I(q) ← I(p)
                    b(q) ← false, λ+(q) ← 0, λ−(q) ← 0


function propagate(p, λ, n_f, n_b)
        λ(p) ← 0
        enqueue in Q_q all neighbors p_f = p + n_f for which b(p_f) = true
        for every element q in Q_q :
            l ← ⋁_i λ(q + n_b(i)) + 1
            if l < λ(q):
                λ(q) ← l
                enqueue in Q_q all neighbors q_f = q + nf for which b(q_f) = true
                enqueue q in Q_c
```

Figure 3.5: Cris Luengo's dimensionality independent path openings algorithm

propagate downstream and upstream length change in the image. Each of the downstream and upstream length values (respectively λ^- and λ^+) is stored in an image and is used to accumulate path length for each of the pixels for the upstream and the downstream direction. Upstream and downstream direction are given by the adjacency relation : for example in the case of a 2D North-South path, the upstream neighbors are {North,North-East, North-West}. The downstream direction is defined by the opposite direction {South, South-East,South-West}.

Initially, for all pixels, λ^+ and λ^- are set to the target path length value L and all the pixels are set to active, meaning that they are all part of a path of length L at the lowest grey level. Starting from an active "seed" pixel, two propagation passes

are made sequentially to update λ^- and λ^+ in the image by enqueuing upstream and downstream active neighbors iteratively. A scheme that outlines the path propagation principle can be found Figure 3.6.

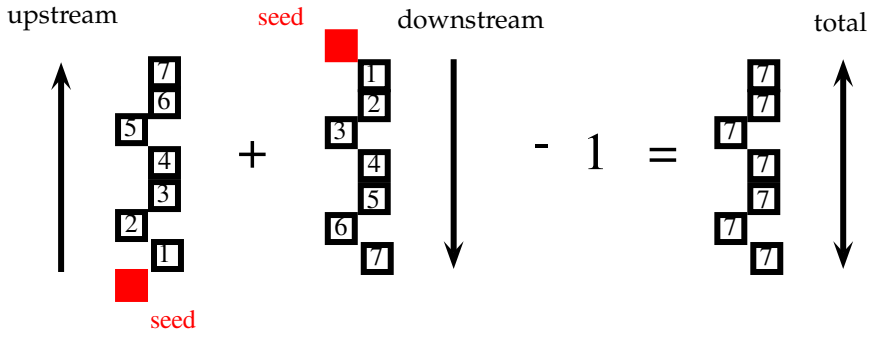


Figure 3.6: Illustration of PO path propagation principle

To enqueue temporarily pixels during the propagation passes, a FIFO queue Q_q is used. For example, in the upstream pass, starting from an active "seed" s, for each of the upstream pixel p in the queue, the maximum length of its downstream neighbors q is found and increased by one :

$$\lambda^- = 1 + \mathrm{MAX}(\lambda^-(SW(q), S(q), SE(q))) \tag{3.11}$$

If λ^- is smaller than $\lambda^-(p)$, λ^- is assigned to $\lambda^-(p)$ and its upstream active neighbors are enqueued. As the target length was set to L, the propagation would stop after a maximum of L iterations or when there would not be any active neighbors to enqueue anymore. During a propagation pass, all the pixels whose downstream value changed are enqueued in a FIFO queue Q_c. After the upstream pass, the downstream pass is done in the same way as for the upstream pass but considering the opposite neighbourhood relationship. When the two passes are over, the maximum length of a path which is going through a pixel stored in Q_c is computed as :

$$\lambda = \lambda^+ + \lambda^- - 1 \tag{3.12}$$

If λ falls under the target value L, this means that this pixel is not part of a path of length L. Its output value is set to the current selected "seed" pixel value. Moreover, as it will no be a part of a path of length L further during the process, its status is set

to inactive and its downstream and upstream length are set to 0.
Note that to render this algorithm suitable for processing n-d images, the implementation starts by creating a linear array of pixel memory adresses. From a given pixel, accessing to the desired neighbors is simply performed by adding the corresponding memory address offset. To simply constrain the path to the image domain, a dark border around the image is added in the opening (resp. light border for closing).
This algorithm is slightly more time consuming than the Talbot and Appleton version, however, it can be readily extended to higher dimensions, and also can be extended to provide more robust operators, as we will see further in this dissertation.
Generally when performing a path openings on a 2D image, four different global orientations are used on which the supremum is performed. Considering a 3D image, we have to adapt the adjacency graph according to the 3D connectivity. We have chosen to use subsets of the 26-neighbourhood connectivity to generate the path thus enabling the computation on 13 orientations. In practice we only use 7 of the 13 possible global orientations because it is sufficient to reach a good anisotropy level in the detection. Figure 3.7 gives some examples adjacency relations used to compute path openings in 3D.

3.6 Limitations Of The Complete Version

Complete path openings and closings described above provide a very efficient tool to detect elongated, narrow and non-necessarily locally straight features. Moreover an efficient dimensionality-independent implementation based on the ordered version by Talbot and Appleton increases the interest that one should give to these operators.
However, complete path operators are very sensitive to noise as it can induce some disconnections along the path (see Figure 3.8). These disconnections prevent the path from being propagated and would lead to an underestimation of the length of the features, thus to their removal (in this context features of interest might be composed of multiple shorter broken paths).

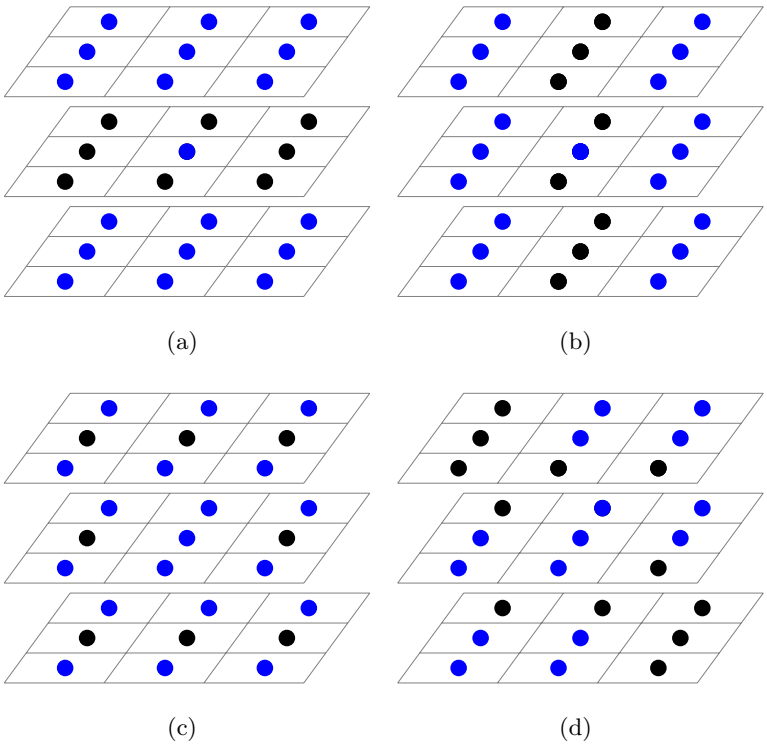


Figure 3.7: Possible adjacency relations in 3D; (a), (b) and (c) are along image axis, (d) is an illustration of a diagonal adjacency relation

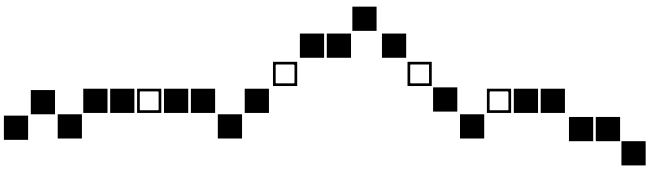


Figure 3.8: Example of a binary feature (black pixels) corrupted by noise (white pixels) generating disconnections that prevent the path from being propagated.

3.7 Incomplete Path Operators

In practical applications, the length of the path L should be increased when one wants to increase the discriminatory power of the path operators relative to compact features or noise. However the longer the path to be detected is, the more chance the path has to contain disconnections, thus the more chance we have to suppress features of interest

(see Figure 3.9 for an example on a retinal vessel image).

3.7.1 Definition

Following the principle of rank max operators (62), (66) incomplete path operators (82) are designed to increase flexibility and noise robustness by allowing an arbitrary number of K missing pixels along a δ-path of length L.
They are particularly suitable for practical applications as they utilize a larger family of structuring elements better able to cope with noisy image features.
An incomplete path openings of parameter K and of length L considers a collection of paths with at least $L - K$ elements inside X. Considering X^c as the complementary set of X in E we have:

$$\Pi_L^K(X) = \{\boldsymbol{a} \in \Pi_L : |\sigma(\boldsymbol{a}) \cap X^c| \leq K\} \tag{3.13}$$

Where $|.|$ represents cardinality, i.e. number of points. From this definition follows immediately the incomplete path opening which is the union of all path contained in $\Pi_L^K(X)$.
For $0 \leq K \leq L$

$$\alpha_L^K(X) = \bigvee\{\sigma(\boldsymbol{a}) \cap X : \boldsymbol{a} \in \Pi_L^K(X)\} \tag{3.14}$$

3.7.2 Implementation And Limitations

In (83), based on the complete path openings implementation, the authors have also provided an ordered implementation for the incomplete path operators.
However, there are some remaining limitations with this implementation. Firstly it requires a significant amount of extra memory and complexity for path bookkeeping, which increases with the dimensionality. So their implementation represents a significant challenge in 3D and up. To the best of our knowledge, such an implementation has not been proposed yet.
Also, the parameter K is typically an increasing function of L. Indeed, as L increases, so does the probability of encountering a noise pixel, so K must increase also to retain effectiveness. However this impacts negatively the algorithm's computational and memory complexity. In the 2D case, it has been shown experimentally that the complexity involved for the incomplete version is $10 \times K$ and the memory requirement is $2 \times K$ that of the complete version.

(a) (b) (c)

Figure 3.9: 2D vessels detection (a).Anisotropic top hat operator applied on the green channel of a retinal image; (b).Result of complete ordered path openings ($L = 45$); (c).Result of incomplete ordered path openings ($L = 45$,$K = 2$); For a better visualization a threshold applied to (a), (b) and (c) (the same threshold value was used);

4

Robust Path Operators

4.1 Motivations and Specifications

Even if a lot of technical advances were realized in order to improve 3D imaging acquisition systems (i.e. acquisition and reconstruction at micro-scale), the reconstructed 3D volumes are often noisy and their automatic analysis often reveals to be a great challenge (e.g. glass fibers reinforced polymer are often at the limit of the resolution of the imaging system).
Path openings and closings have shown their efficiency for the detection of thin, curvilinear and non-necessarily straight features. In the context of real data corrupted by different sorts of noises (e.g. noise coming from acquisition and/or reconstruction), we will propose a new framework able to handle in an original manner the disconnections affecting the propagation of the path.
Nevertheless, some constraints are imposed to our work:

- the developed algorithm should be working for both 2D and 3D images,
- the complexity should be reduced,
- the memory requirement should be constant,
- the parameter handling the robustness of the path should not be dependent on the length of the path.

4.2 Proposed Solution

A logical solution to the filtering of thin features in 3D should be to provide a d-dimensional implementation of incomplete path openings. However we have seen that the complexity involved while running this algorithm on 3D images should be prohibitive. Indeed what make the complexity of incomplete path openings described earlier in this dissertation is the fact that the robustness aspect is handled during the propagation of the path. Indeed, for each of the missing pixel, the algorithm needs another temporary image to "remember" the fact that it went through a "noise" pixel. Keeping in mind that the path should go through "noise" pixels (i.e. disconnections) we propose another framework to handle its robustness.
We have designed a d-dimensional algorithm, the robust path openings (85), based on the dimensionality-independent path openings implementation of (81). For each pair of parameters (L,G), a larger family of flexible and incomplete structuring elements than the incomplete path operator is generated.
Given an arbitrary length G, path propagation is allowed if the disconnection length between two elements of an incomplete path is less or equal to G. This renders G independent of L. This simple assumption leads to the generation of a more flexible and more efficient family of structuring elements, due to constant memory bookkeeping requirements and a lower-complexity implementation.

4.3 Theoretical Aspects

We will state in this section the theoretical aspects of the robust path openings.

4.3.1 Principle

Given an arbitrary length G, path propagation is allowed if the disconnection length between two elements of the path is less or equal to G ($1 \leq G < L$).
Pixels belonging to this disconnections will be termed *noise* pixels.

4.3.2 Searching for Noise Pixels

Noise pixels can be defined as being part of a path in X^c (i.e. in the background) of maximal length G pixels in which first point and end point are in $\delta(X)$ and $\breve{\delta}(X)$

respectively.

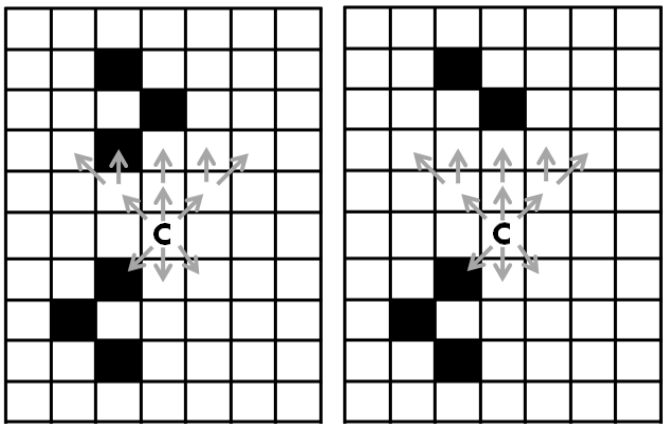
Now we can define a *noise path condition* NPC_k meaning that a δ-path of $\Pi_k(X^c)$ with $1 \leq k < G$ must have at least one of its first element's predecessors and at least one of its last element's successors in X. For $k = 1...G$ we have for a path $\boldsymbol{a}$ of length k in X^c (Figure 4.1 gives an illustration for $G = 2$):

$$\mathrm{NPC}_k(\boldsymbol{a}) = \begin{cases} \check{\delta}(\{a_1\}) \cap X \neq \emptyset \\ \delta(\{a_k\}) \cap X \neq \emptyset \end{cases} \tag{4.1}$$

a_1 is the first point of the path and a_k is the last point of the path in X^c.

So we define the set $\Pi^G_{\mathrm{noise}}(X)$ as :

$$\Pi^G_{\mathrm{noise}}(X) = \{\boldsymbol{a} \in \bigvee_{k=1..G} \Pi_k(X^c) : \boldsymbol{a} \;\; satisfy \;\; \mathrm{NPC}_k\} \tag{4.2}$$



(a) (b)

Figure 4.1: Noise pixels search for $G = 2$ (a).The candidate **C** is flagged; (b).The candidate **C** is not flagged.

4.3.3 G-robust paths

Here we define the set of G-robust δ-path of length L For $1 \leq G < L$

$$\Pi^G_L(X)_{\mathrm{Robust}} = \Pi_L(X \cup X^G_{\mathrm{noise}}) \tag{4.3}$$

with

$$X^G_{\text{noise}} = \{x \in \sigma(\boldsymbol{a}_{\text{noise}}) : \boldsymbol{a}_{\text{noise}} \in \Pi^G_{\text{noise}}(X)\} \tag{4.4}$$

$\sigma(\boldsymbol{a}_{\text{noise}})$ is analogously defined as being the set of points belonging to a path of $\Pi^G_{\text{noise}}(X)$. So we define the robust path openings as:

$$\alpha^G_L(X)_{\text{Robust}} = \bigvee\{\sigma(\boldsymbol{a}) \cap X : \boldsymbol{a} \in \Pi^G_L(X)_{\text{Robust}}\} \tag{4.5}$$

Roughly speaking, robust path openings are equivalent to complete path openings considering an extended propagation set. Anti-extensivity and idempotence properties are straightforward, however we should prove the increasingness property.
Basically, complete path openings are a morphological filter that respects the increasingness property (1). To prove that of robust path openings, as it relies on complete path openings principle, we should prove that the operator giving a 'robust propagation set' from an arbitrary given set is increasing too.

Proof:
Here we place ourselves in the set of all subsets of E, $\mathcal{P}(E)$.
Considering X and Y of E (on Figure 4.1, X could be (b) and Y could be (a)) with $X \subseteq Y$, the robust set of X is $X \cup X^G_{\text{noise}}$ ($Y \cup Y^G_{\text{noise}}$ is defined analogously). Basically we can state that we have $X^G_{\text{noise}} \subseteq Y^G_{\text{noise}} \cup (Y \setminus X)$.
Thus we obtain $(X \cup X^G_{\text{noise}}) \subseteq (X \cup Y^G_{\text{noise}} \cup (Y \setminus X))$ and then $(X \cup X^G_{\text{noise}}) \subseteq (Y \cup Y^G_{\text{noise}})$
One can notice that the operator giving a robust propagation set from an arbitrary subset of E as the property of a dilation (specifically increasingness, extensivity and commutativity with the *supremum*).
Now we can state that robust path openings (resp. closings) have all the algebraic properties of an opening (resp. a closing): increasingness, idempotence and anti-extensivity (resp. extensivity). Allowing paths to admit noise pixels simply means that we admit a larger family of structuring elements over which we perform a supremum (resp. an infimum).

4.4 Implementation Issues

An overview of our proposed algorithm for robust path openings is outlined in the pseudo-code Figure 4.2 and a scheme can be found Figure 4.3. A more complete form

Figure 4.2: Overview of d-dimensional robust path openings.

- Initialisation :
  - Initialise the three temporary images needed for the processing :
    - *Downstream* and *usptream* images are initialised to target path length value L
    - All the pixels are set to active, all the others flags are reset
    - Border pixels are prevented from being flagged as *noise*, their *upstream* and *downstream* lengths are reset
  - Sort pixels memory addresses list by value (low to high for opening)
  - Compute memory address offset for each of the neighbor pixel (given by the adjacency relation)
- Path propagation :
  - For every pixel p that :
    - is not flagged as *noise* pixel
    - can be considered as a "seed"
  - Determine the current threshold $T = I(p)$
  - If T changes, then deactivate all *active* pixels whose value equal to T then :
    - Determine *noise* pixels within the previous deactivated set : enqueue *noise* pixels in Q_N (upstream and downstream lengths are not reset)
    - Update *noise* pixels enqueued in $Q_{N_{\text{check}}}$: enqueue pixels that are still *noise* pixels in Q_N, those which are not in $Q_{N_{\text{propagation}}}$
    - Empty Q_N into $Q_{N_{\text{check}}}$
    - Make *upstream* and *downstream* passes for pixels in $Q_{N_{\text{propagation}}}$, store pixels whose length changed in Q_c
    - Update the status of pixels enqueued in Q_c
  - Make *upstream* and *downstream* passes for p, store pixels whose length changed in Q_c
  - Update the status of pixels enqueued in Q_c

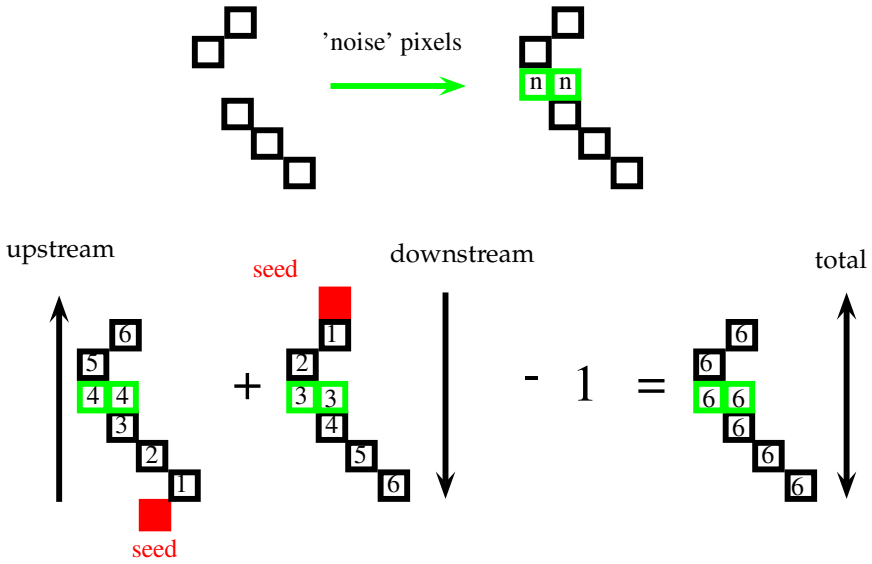


Figure 4.3: Illustration of RPO path propagation principle

of this algorithm can be found at the end of this chapter.

4.4.1 Principle

At each threshold change, given a length G, the set of *active* pixels is enlarged with the pixels satisfying a specific condition : they belong to a path of maximum length G formed by *inactive* pixels. These pixels are flagged as *noise* and their *upstream* and *downstream* lengths are not reset. During the propagation of the path, *noise* pixels act like *active* pixels and will prevent the propagation from being interrupted by small disconnections. Obviously, in the greyscale case, at each threshold change we have to update the set of *noise* pixels from previous thresholds considering the newly deactivated pixels (this will ensure that robust path openings will respect the stacking property).

After the updating procedure, a *noise* pixel which is no longer a part of a path of maximum length G formed by *deactived* pixels is enqueued is $Q_{N_{propagation}}$ to update length change in the image, otherwise its status and lengths are not changed.

4.4.2 Description of Flags

As seen above, we need to add a "status" flag to handle noise pixels (N). For efficiency reasons, two other flags are added to prevent active pixels that are deactivated during the propagation of the path to, enter at next threshold into the noise pixels research procedure ($\overline{N}$) (as $G < L$ this pixel cannot be flagged as a *noise* pixel at the next threshold) and, to prevent from being considered as a seed again ($\overline{s}$). This is due to the fact that this pixel will not be a part of a path of length L further in the program.

4.4.3 Determine Noise Pixels

The search procedure of *noise* pixels is performed at each threshold change when :

- we determine *noise* pixels from the newly deactivated pixels set,
- we update the set of previous *noise* pixels.

Given an arbitrary length G, from a candidate *noise* pixel, two propagation passes are made from $\mathbf{C}$ (in *upstream* and *downstream* direction) seeking *active* pixels in the range of a maximum of G iterations. We define it_{up} and it_{down} as respectively the minimum of *upstream* and *downstream* iterations necessary to find an *active* pixel from $\mathbf{C}$. If $it_{\text{up}} + it_{\text{down}} - 1 \leq G$, we flag $\mathbf{C}$ as *noise* pixel and enqueue it in Q_N, otherwise this pixel cannot be a *noise* pixel and have to be used as a seed to propagate the change in lengths (its corresponding λ^+ and λ^- are set to 0).
Obviously, as long as its status is set to *noise*, a pixel cannot be considered as a seed (because it belongs to the propagation set). Figure 4.1 illustrates two possible cases for $G = 2$. In a), we can find an active pixel (in black) making G iterations considering the *upstream* and *downstream* directions. We have $it_{\text{up}} = 2$ and $it_{\text{down}} = 1$, this pixel is flagged as *noise*. In b), we have $it_{\text{up}} > 2$ and $it_{\text{down}} = 1$, this pixel cannot be flagged as a *noise* pixel.

4.4.4 Robust Propagation

Now that we have extended the set of propagation pixels, we enter into the path propagation function following the same principle as in dimensionality independent path openings (81). *Noise* pixels are enqueued in Q_q in the same way as *active* pixels are enqueued. Figure 4.14 outlines the robust propagation algorithm. Considering both

active and *noise* pixels, at the end of the two propagation passes, each of the pixels whose λ^+ or λ^- value changes is enqueued in Q_c.
For each of the pixels in Q_c the length of the longest path passing through it is computed as we have seen previously in the explanation of dimensionality independent path openings algorithm. For an *active* pixel q, if this value falls under L, its corresponding *upstream* and *downstream* lengths are set to 0. Its status is set to *inactive*. Obviously, this pixel will neither be considered as a "seed" nor as a *noise* pixel (flags $\overline{s}(q) \leftarrow true$, $\overline{N}(q) \leftarrow true$).
For a noise pixel q, if this value falls under L, its corresponding *upstream* and *downstream* lengths are set to 0 (to prevent this pixel to be considered as a seed in the current threshold its flag $\overline{s}(q)$ is set to $true$). As in Luengo's algorithm it is possible to write directly in the input linear buffer, an additional output buffer is not needed.

4.4.5 Border Issue

As with dimensionality independent path openings algorithm, we add a dark border (resp. bright border for closing) around the image to constrain the path to the image domain. At the initialisation step, to prevent these pixels to enter into the noise pixels research fonction we set their $\overline{N}$ flags to true. Their corresponding lengths λ^+ and λ^- are set to 0.

4.4.6 Path Reconstruction

With no extra cost in execution time, we can modify in a simple way the robust path operator to reconstruct disconnections that affect thin features. The idea is to enhance *noise* pixels to the highest threshold value for which the binary robust path operator remains true in order to fill the gaps between two *active* elements of the path (see pseudo-code in Figure 4.2). In the case of a robust path opening (resp. closing), with respect to the result of the opening, the result of this reconstruction operator forms a closing (resp. opening). Indeed it is clearly extensive (resp. anti-extensive), it is also increasing and idempotent.

4.5 Experimental Study

In this section, we evaluate the benefits of using robust path openings for the detection of thin human vessels in Magnetic Resonance Angiography (MRA) using the simulated data set presented in (86). The vessel length and radius are respectively 90 and 2 pixels.

We use a classical evaluation method proposed in (87) and (88), which consists in generating a noisy simulated image by the addition of white Gaussian noise on a known ground truth image. Background is set at a grey level of 100, the vessel at grey level of 160 which are typical values of MRA acquisition. Note that, as structuring elements generated by the path openings are one pixel thick, they can also detect vessels of smaller or larger widths.

We then added white Gaussian noise with standard deviation σ equal to 35 to test the performance of our method (typical MRA noise levels are more around $\sigma = 20$). We filtered this image with both Complete and Robust ($G = 1$) path openings of length equal to 90 in the z-axis direction.

Figure 4.4 shows 3D renderings of ground truth, input and result images using the maximum intensity projection method. We can note that disconnections due to noise causes the complete path opening to fail to detect the vessel (orange color for the vessel (c)) whereas the robust path opening provides a higher detection rate (red color for the vessel (d)). As could be expected, the noise is greatly reduced by the complete path opening (green background) but contrast in the feature is also greatly reduced. The robust path opening preserves the feature much better but also some of the noise compared with the complete path opening. This phenomenon results in a higher false alarm rate (yellow color for the background (d)).

To cope with this limitation in a simple fashion, we can apply a shorter complete path openings ($L = 10$ in the same orientation) on the result of the robust path openings to eliminate the smaller residuals. Figure 4.4 (e) shows that we are able to detect the feature of interest and to simultaneously reduce significantly the amount of false positives (on this image, the feature is in red and background appears in green). Note that the same colormap is used for images (b) to (e).

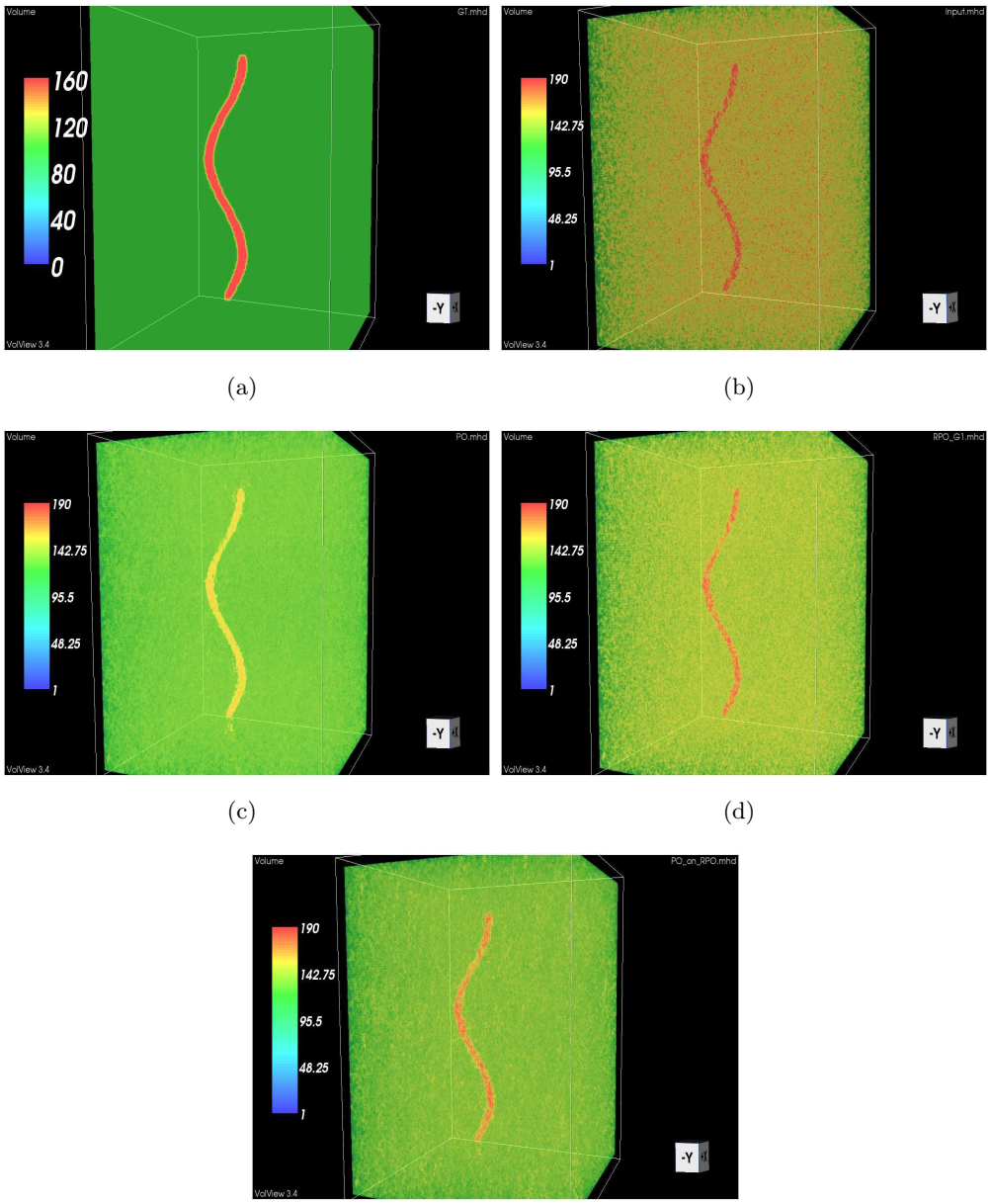


Figure 4.4: $3D$ rendering of (a).Ground truth image; (b).Noisy input image($\sigma = 35$); (c).Result of path openings ($L = 90$); (d).Result of robust path openings ($L = 90$, $G = 1$); (e).Path openings ($L = 10$) applied on (d);The same color mapping was applied for (b), (c), (d) and (e); image is 128^3x8-bit

4.6 Comparisons with Complete and Incomplete Versions

Here we present results regarding to the regular path opening, the incomplete version and the robust version.

Improving the detection of narrow and low-contrast features while reducing the false positive detections is the goal of vessel extraction in a retinal image. This problem has been widely considered in the past few years (89), (90).

Firstly the green channel of the color retinal image is extracted to obtain a grey-scale image, as the green channel provides the best signal to noise ratio. Secondly a square top-hat filter is applied on the image to extract the local minima of the image. The top-hat filter extracts the vessels but generates a large amount of noise. We compare the results of complete, incomplete and robust path openings on the top-hat image, performing the union of the four orientations in each case.

Figure 4.5 shows the thresholded version of all the results on a region of interest (the region of interest was chosen where the vessels are thin and low contrasted). Using complete path openings, the totality of the noise is eliminated, however, we notice that this operator is not able to recover the vessels presenting some disconnections. Incomplete path openings give better results, the vessels are better recovered but some vessels which are very affected by the disconnections are not (if the number of gaps in a path of length L is greater than K, path propagation is stopped). We can note also the generation of some artifacts due to the linkage of some random noise pixels together. Robust path openings visibly provide the best results for the detection, as some vessels which are not present in the incomplete path openings result were recovered. However, as we are increasing the tolerance of the path opening operators, we generate a larger amount of artifacts.

Figure 4.6 compares the running times of the different algorithms using the top-hat retinal image as input. As expected, the complexity of the robust path openings is lower compared to incomplete path openings in all cases. For example, for $G = 2$ and $K = 2$, robust path openings is 7 times faster than incomplete path openings. Moreover, robust path openings has constant memory requirements irrespective of the parameter G.

We observe that, given a fixed L, the running time of robust path openings is linear with G (about $0.4\times$G). We can also note that the parameter L does not affect computation

time much for given a fixed G, indeed, we believe that it is regulated between *noise* pixels research and path propagation.

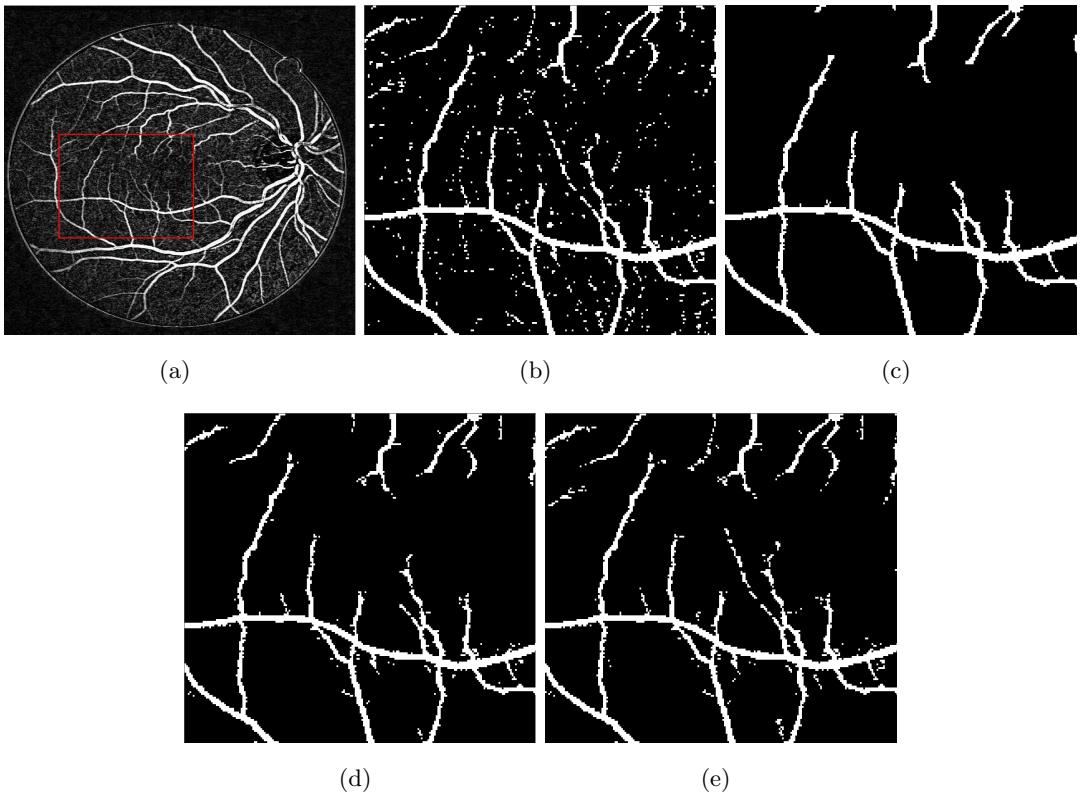


Figure 4.5: 2D vessels detection (a).input image : top-hat filter applied on the green channel of a retinal image, the red rectangle represents the region of interest used for comparisons of the different algorithms' results; (b). Thresholded version of the input image; (b). Result of complete ordered path openings ($L = 45$); (c). Result of incomplete ordered path openings ($L = 45$, $K = 2$); (d). Result of robust path openings ($L = 45$, $G = 2$); the same threshold was applied for (b),(c),(d) and (e) for comparable visualizations.

4.6.1 3D Data

Here we give an example for the use of robust paths for biomedical imaging on 3D data. We propose to process a $3D$ cerebral angiogram image in order to detect the network of vessels while removing the noise. To do this, we apply both complete and robust path openings of length $L = 15$, performing the union of three orientations (orientations are in the x, y and z axis with a 9-neighbours adjacency relation).

	COPO	RPO	IOPO	RPO	IOPO	RPO	IOPO
Tolerance	$K = 0$	$G = 1$	$K = 1$	$G = 2$	$K = 2$	$G = 3$	$K = 3$
L=10	0.71	1.03	5.61	1.34	8.78	1.81	12
L=50	0.79	1.03	5.93	1.33	9.31	1.77	12.93
L=100	0.82	1.05	6.03	1.35	9.47	1.78	13.2

Figure 4.6: Execution times of Complete Ordered Path Openings (COPO), Incomplete Ordered Path Openings (IOPO) and Robust Path Openings (RPO) with different tolerance parameters (K and G). Timings are in seconds. Image is 565x585x8-bit.

Figure 4.7 shows that the robust operators have the ability to recover a larger part of the vessels (e.g. the part of the vessel in the red circle) compared to the complete ones while eliminating most of the noise. The running times of complete and robust ($G = 1$) grey-scale path openings are respectively 10.5 and 18.6 seconds (45 and 124 seconds for $G = 2$ and $G = 3$ respectively). The image dimensions were 256×256×256×8-bit. In this case, our implementation achieves quite good performance since the background is flat. For a non-flat background (in the worst case, for a white Gaussian background) our implementation could result in higher running times. Figure 5.10 shows that the robust path openings is able to both detect and then reconstruct the thin and corrupted structures.

As noise pixels are handled like active pixels during the propagation of the path, we can make use of the status change of noise pixels to give them the highest value for which they belong to a path. This is very efficient in practice as it would be possible, for the practitioner in medical imaging for example, in vascular network analysis applications, to recover some information that is not present usually after a thinning operation.

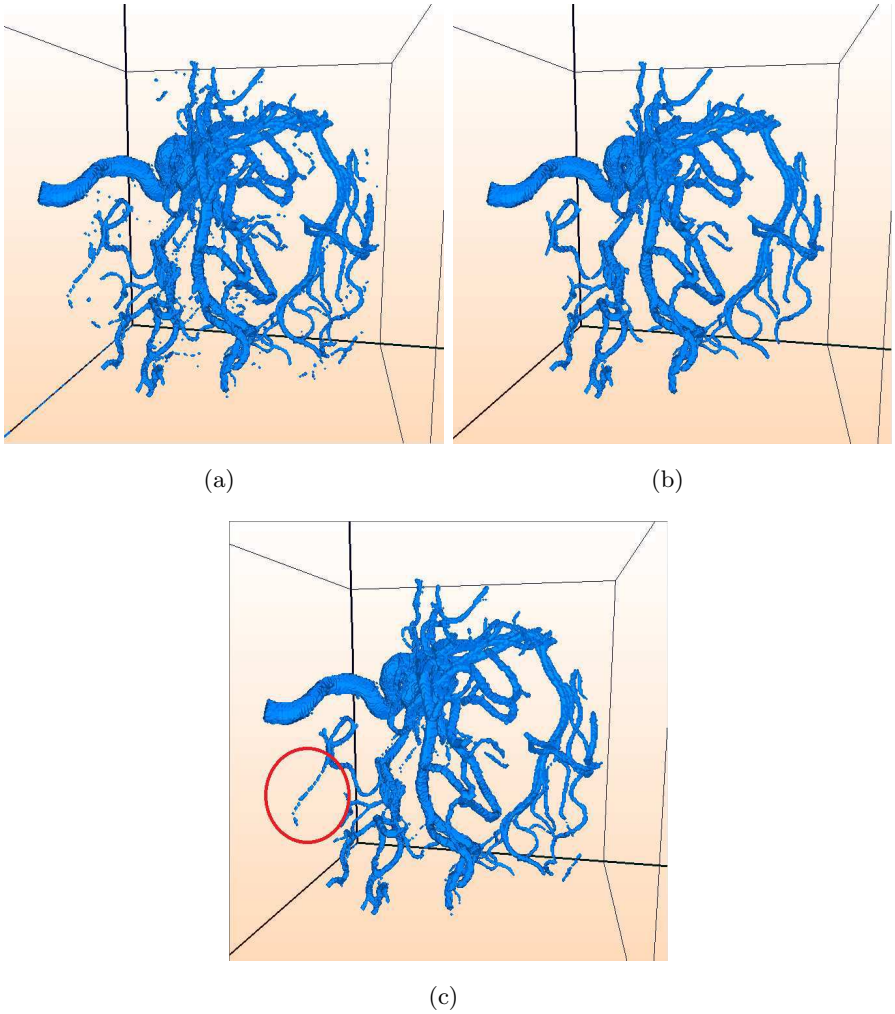
4.6.2 Quantitative Evaluation

4.6.2.1 How to Evaluate an Image Processing Algorithm ?

Even if it is relevant to compare qualitatively the results of the different algorithms, a quantitative comparison is needed to ensure their efficiency.

There are mainly three ways to make an evaluation of an image processing algorithm:

- using synthetic images wherein the rate of noise is controlled,



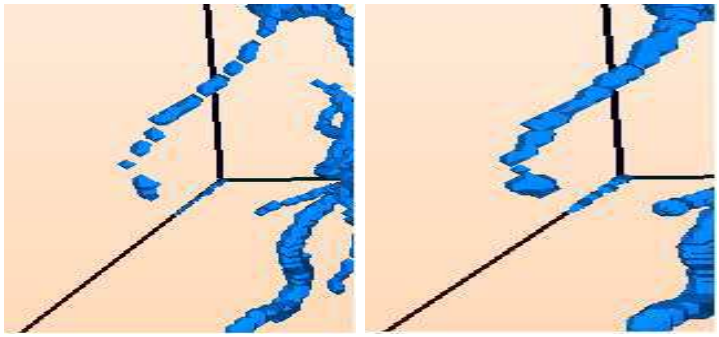
(a) (b)

(c)

Figure 4.7: Surface rendering of (a).input noisy angiogram; (b).complete path openings; (c).robust path openings ($G = 1$); Binarisation is performed by choosing a threshold manually.

- building a physical phantom with known characteristics, making a real acquisition and then compare the computed characteristics,
- having a ground truth generated from a real acquisition which was analyzed and segmented by an expert.

The third method will be used in this work. Obviously, the results will depend on the application chosen to compare the different methods.



(a) (b)

Figure 4.8: Surface rendering of (a).robust path openings with no reconstruction ($G = 1$); (b).robust path openings with reconstruction ($G = 1$);

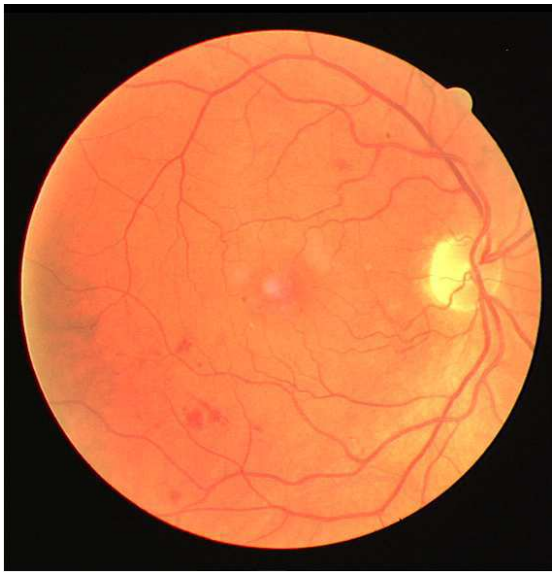
Here we propose to perform an evaluation on the well-known DRIVE database (91) containing more than 40 of retinal images and their corresponding binary ground truth. As an illustration, an image from the set of test images has been randomly chosen (see Figure 4.9). Note that this image presents curvilinear vessels of various sizes and compact dark features.

Note also that we are performing this evaluation in 2D mostly for practical reasons. Indeed, most of the implementations that can be found are often in the 2D version. However, most of them have their theoretical equivalent in 3D and so we believe that these results can be reasonably extended to the 3D case.

4.6.2.2 Context of this Experimental Study

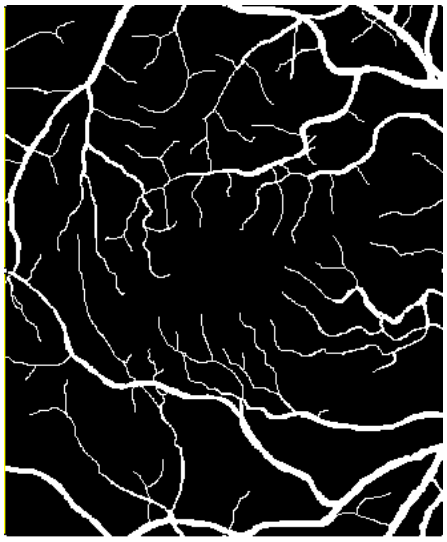
In order to get specific attributes of retinal blood vessels such as inherent geometrical characteristics (length, width, elongation, tortuosity...) or network connectivity, several methods were developed to automate their extraction (mostly because manual segmentation reveals to be time consuming and subjective to the operator).

Vessel segmentation is critical for the practitioner in biomedical imaging in order to diagnosis various diseases especially cardiovascular and ophtalmologic ones (see (92) for a recent review). To reach an acceptable automated segmentation, preprocessing steps aiming at enhancing vessels from the background are essential. This task is directly linked to the detection of narrow and curvilinear features in a low contrast image.



(a)

(b)



(c)

Figure 4.9: Retinal image used in this evaluation: (a).Original color image; (b).After extracting a ROI (Region Of Interest) and the green channel of the color image; (c).Provided ground truth image manually segmented by an expert

In this section we aim to evaluate the suitability of the robust path operators and other well known state of art's methods (presented in the first chapter of this dissertation)

for the generation of filtered versions of the input retinal image that can be further considered for the extraction of the vessels. The evaluation has been performed by considering the DRIVE database (91) and by assessing the accuracy with respect to the available ground truth of simple binary segmentation maps obtained by thresholding the output of the filters.
The performances were quantitatively assessed by computing Receiving Operator Curves (ROCs) and by evaluating the accuracy of the segmentation map correspondent to the Optimum ROC Point (opt) curve. An additional contribution of this work is the proposal of an efficient preprocessing pipeline (see Figure 4.10) aiming at enhancing the vessels starting from a RGB retinal image and generating an input image for all the methods used for comparisons.

Preprocessing pipeline:

Starting from the RGB image acquired by the opthalmologic camera, the preprocessing

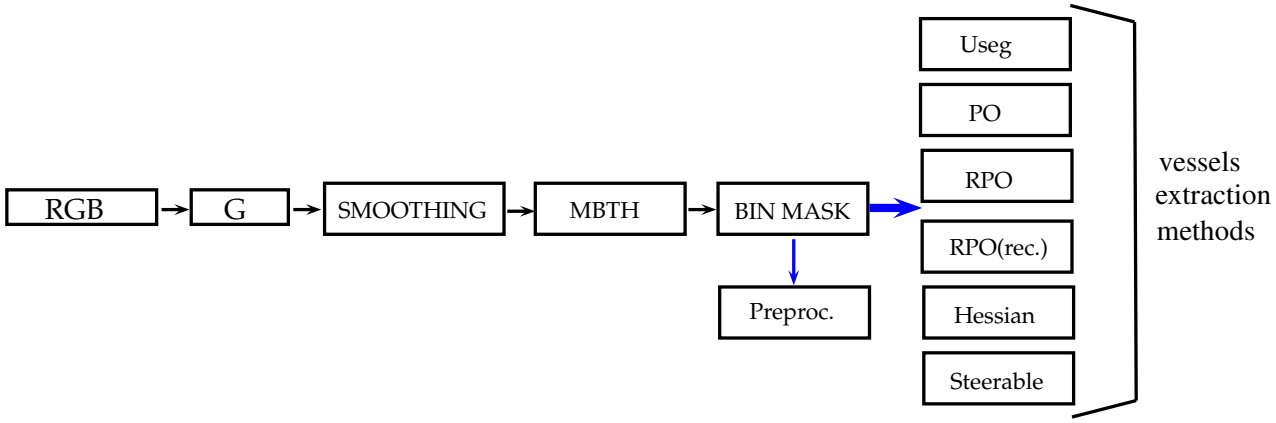


Figure 4.10: Preprocessing pipeline providing the input image for all the methods.

pipeline aims at preparing the image for the subsequent application of the filters. The preprocessing operation enhances the vessels by removing the noise and suppressing the heterogeneously contrasted background. Classically the green channel of the RGB image is extracted as it provides the best contrast between the vessels and the background (92). A smoothing step is then performed by using a bilateral filter based on Gaussian kernels with parameters $\sigma_d = 2$ and $\sigma_r = 3$. This filter is used *in lieu* of classical isotropic smoothing filters to cope with the blurring effect affecting the thin vessels. Then, an isotropic modified black top hat (MBTH) is applied on the image. Basically a regular black top-hat operator (BTH) is retrieving all the features smaller than B the structuring element (SE) and darker than their surrounding ($BTH(X) = \phi_B(X) - X$, with $\phi_B(X)$ the closing of X using B as a SE). Since the result of the BTH enhances the dark structures associated to vessels but also small contrast variations in the background, we applied a MBTH, which is defined as:

$$MBTH(X) = \max(\phi_{B_1}(\gamma_{B_2}(X)), X) - X, \tag{4.6}$$

with $\gamma_B(X)$ the opening of X using B as a SE. The result of this operator should be close to zero except for large dark structures (e.g., vessels).

Here we have used SEs with disks shapes B_1 and B_2 of radius 4 and 1 respectively. Selecting a small size for B_1 ensures that all the vessels will be retrieved (as they are

about 8 pixels of width). The size of B_2 was selected in order to be larger than the maximum expected width of the vessels. A binary mask (available in the DRIVE database) is then applied to the MBTH image in order to perform the vessels extraction methods only on the area of the image showing the retina.

Even if the methods used for comparisons have been presented in the first chapter, we will remind briefly their principles and the parameters settings for each of them.

Methods Used for Comparisons Multi-scale filter:

Considering a 2D image as an elevation map of $\mathbb{Z}^2 \rightarrow \mathbb{R}$, derivatives methods are assuming that features of interest represent the crest lines of this elevation map (in the case of the detection of brighter features). Using kernels based on the widely used derivatives of Gaussian, one can access to the differential property of the image in several directions and then detect features presenting particular shapes (e.g. tubular objects for example). Hessian filters (8, 14), are one of the most widely used transformations in the community of medical imaging. The benefit of using the Hessian filter is mostly relying on its property of being multi-scale. By accounting for different scales it is possible to retrieve both smaller and larger features. Hessian based filters are obtained by convolving the image with second order derivatives of Gaussian kernels with different standard deviations σ thus enabling the characterization of the local geometry of a pixel according to a given scale.
After building the Hessian matrix and extracting its two eigenvalues ($\lambda_{1,2}$), a function ν (called vesselness function) can be used as a measure of the likelihood of each pixel to belong to a vessel. In this study we have used the Frangi measure (8), which for a pixel $p \in \mathbb{Z}^2$ is defined as:

$$\nu_{max}(p) = \max_{\sigma_{range}} (\nu(p, \sigma)), \tag{4.7}$$

with σ_{range} the range of ridges widths that are expected to be found in the image.
Here we have used the implementation available as a plug-in based on the ITK image processing library (14). Also, a multi-scale approach was used considering $\sigma_{range} = [1, 4]$.

Oriented filters: Oriented filtering is realized by performing a matched filtering considering a rotated version of an arbitrary chosen basis kernel (e.g., Gaussian kernel are

the most used). The finality is to examine the response of the filter at many orientations (by rotating the basis kernel in several orientations) and to get the greatest one. Steerable filters (5, 6) are a subclass of oriented filters in which the response at an arbitrary angle can be computed efficiently and analytically as a linear combination of a set of basis filters to reduce the computational time of a direct implementation (i.e. the implementation of each of the rotated version of the basis kernel).

Derivatives of the Gaussian kernel of all orders have been proven to be steerable and very efficient at detecting ridges in images. Here we have used the implementation provided by (6) as a plug-in for ImageJ software (93) and we set the Gaussian kernel width as $\sigma = 2$.

	$\bigcup$segments	PO	RPO	RPO (recons.)	Hessian	Steerable	Preproc
FPR_{opt}	$\mathbf{0.0104 \pm 0.0024}$	0.0115 ± 0.0031	0.0114 ± 0.0029	0.0175 ± 0.0040	0.0140 ± 0.0019	0.0111 ± 0.0013	0.0108 ± 0.0015
TPR_{opt}	0.6003 ± 0.0521	0.6840 ± 0.0457	0.6867 ± 0.0383	$\mathbf{0.7115 \pm 0.0356}$	0.6759 ± 0.0296	0.6435 ± 0.0356	0.6678 ± 0.0506
$Prec_{opt}$	0.8478 ± 0.0226	0.8519 ± 0.0316	0.8528 ± 0.0340	0.7964 ± 0.0417	0.8223 ± 0.0200	0.8472 ± 0.0182	$\mathbf{0.8554 \pm 0.0200}$
Acc_{opt}	0.9556 ± 0.0043	0.9619 ± 0.0043	$\mathbf{0.9621 \pm 0.0046}$	0.9588 ± 0.0033	0.9590 ± 0.0037	0.9587 ± 0.0048	0.9611 ± 0.0053

Figure 4.11: Accuracy values computed on the ROC curves of the 20 images of the DRIVE database and the manual segmentation of the first observer considered as ground truth. For each image the segmentation map correspondent to the optimal ROC curve point (*opt*) (94) was selected. Averages over all the images and standard deviation values of the False Positive Rate (FPR), True Positive Rate (TPR), Precision (Prec) and Accuracy (Acc) are reported.

Union of line segments: Performing the union of the response of oriented line segments within a large range of orientations (considered as SEs) should be a good answer to the problem of the detection of thin features especially when the orientations of the features in the image are unknown. Soille *et al.* proposed an efficient algorithm invariant to translation to compute erosions and dilations along Bresenham lines (62). A "rank-based" version of this algorithm was also proposed enabling the computation of openings with incomplete oriented line segments (i.e. with a given number of points lying into the background of the image). Given a line segment SE B (with length L and orientation θ) whose cardinality $\mathrm{card}(B)$ equals n points, the rank-max opening with parameter r is equivalent to the union of all the SEs B_i included into B and containing r pixels:

$$\gamma_{B,r} = \bigvee_i \{\gamma_{B_i} \mid B_i \subseteq B \cap \mathrm{card}(B_i) = r\}. \tag{4.8}$$

Compared to the use of plain line segments, rank-max openings are less sensitive to the presence of small gaps (indeed gaps up to $(n - r)$ are allowed along the considered oriented line segment). This reinforces their applicability to the detection of features in noisy environment. Here we have used the following parameters: $L = 30$, $r = 80\%$ and step size for θ of 30^o.

Complete path operators: In the experiments we considered four orientations i.e. horizontal, vertical and two diagonals with $L = 45$.

Robust path operators: We have used $G = 1$ and $L = 45$ considering four orientations (same as the complete version) as it seems to be a good compromised between discriminatory power and radiometric flexibility.

4.6.2.3 Evaluation

The experiments have been carried out on the DRIVE database (20 images) using the 1^{st} observer manual segmentation results as *ground truth* for comparison. Following the evaluation method described in (72), based on the ground truth image, ROC (Receiver Operating Characteristic) (94) curves have been computed to assess the performance of each of the method. We underline once again that the experiments were performed for evaluating the robust path openings efficiency compare to other filtering techniques with respect to the task of vessels detection and that we do not propose a complete vessel extraction technique. From each ROC curve we can extract its optimal point corresponding to the segmentation map that maximizes the TPR for a minimum of FPR (i.e., the point on the curve closest to the left upper corner).

4.6.2.4 Results

The results have been reported in Table 4.11 and the obtained segmentation map for each of the method considering *test image 05* has been reported in Figure 4.12. Due to the first preprocessing pipeline, the results obtained for all the methods (and also for the preprocessing result image) are very good compare to state of art's methods.

Moreover these results show RPO to be one of the best methods compared to the others. Indeed it provides the best accuracy (0.9621) and is second for precision (0.8528). These results can be confirmed visually looking at Figure 4.12. Combining the flexibility of path operators with a radiometric robustness enables to retrieve the smallest and not well contrasted vessels (see the up and right corner of the images)

and eliminate compact noise. We can note also that detecting filters are very efficient but they can give strong responses to non-relevant features. Mathematical morphology methods are useful in the sense that they provide a higher connectivity in vessels' network which is critical for further analysis.

4.7 Discussion

In this chapter, we have defined the robust path operators and we have proposed an efficient algorithm for their computation. We have specified the theoretical foundations of robust path operators.

Our algorithm uses a constant amount of memory and its complexity is proportional to the jump length G ($0.4\times$G as shown experimentally in 2D). We have shown the effectiveness of our algorithm for detecting thin, oriented features in noisy images on both simulated and real 2D and 3D data, and we have compared our results to complete and incomplete path operators.

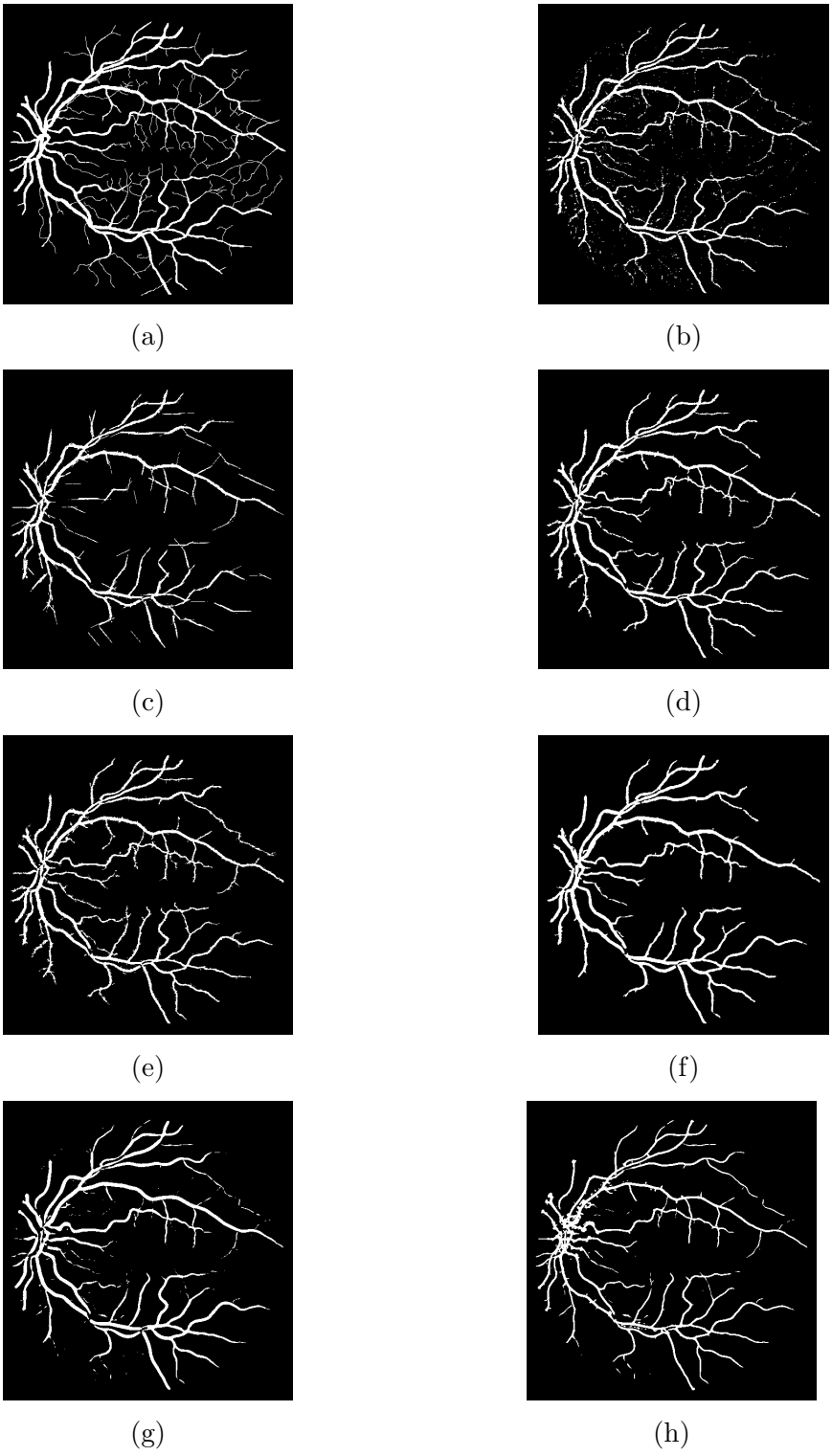
Timings comparisons with path based algorithms were also achieved and have proven the robust path operators to be quite efficient in practice. Although there is no equivalent implementation of incomplete path operators in 3D for robust path operators to compare with, we believe that the 2D results would carry over to 3D as the complexity scales with the number of voxels involved. We have also proposed an extension to robust path openings which can be used to reconstruct the noisy gaps between feature elements.

As the robust path operators increase flexibility significantly, in the case of a White Gaussian background, they may generate a higher false alarm rate because isolated noise pixels may be lumped together with aligned pixels. To cope with this limitation, we have proposed to combine both robust and complete path openings sequentially. In this case, we reach a high detection rate for a lower false positive detections rate.

A quantitative evaluation was also performed in 2D on the DRIVE retinal image database and has shown that robust path openings was one of the best method for the detection of thin, elongated and flexible features in the case of real data compare to 'state of art' methods.

While the implementation of our algorithm is inspired from the dimensionality-independent

path operators, we provide a new framework to cope with disconnections corrupting thin, oriented and flexible noisy features in d-dimensional images.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4.12: Test image 05 of the DRIVE database: (a) ground truth (1^{st} observer); (b) preprocessed image input of the vessels extraction methods, (c) rank-max openings, (d) complete path openings, (e) robust path openings, (f) robust path openings with reconstruction of the noisy gaps, (g) multi-scale Hessian filter based on the Frangi vesselness measure function and (e) steerable filters. For each of the method, the segmentation correspondent to the optimum ROC point is presented.

***d*-Dimensional Robust Path Openings**

create list of offsets n^+ to upstream neighbors and n^- to downstream neighbors
sort pixels' list indices i according to value of $I(i)$
Initialize : $b \leftarrow true$, $N \leftarrow false$, $\lambda^+ \leftarrow L$, $\lambda^- \leftarrow L$
Initialize : $\overline{N}(b_p) \leftarrow true$, $\lambda^+(b_p) \leftarrow 0$, $\lambda^-(b_p) \leftarrow 0$

For every element p in i which $\overline{s}(p) = false$ and $N(p) = false$
    $threshold \leftarrow I(p)$
    If *threshold* change
        //current threshold pixels deactivation
        For every pixel q for which $b(q) \leftarrow true$ and $I(q) \leq threshold$
            $b(q) \leftarrow false$
            If $\lambda^+ + \lambda^- - 1 \geq L$ **Then** $I(q) \leftarrow threshold$
        End For
        //determine noise pixels set from previous deactivation pixels set
        For every pixel q for which $b(q) \leftarrow false$, $\overline{N}(q) \leftarrow false$ and $I(q) \leq threshold$
            If q belongs to a path of maximum length G in the *deactivated* set
                $N(q) \leftarrow true$ *// upstream and downstream values are not changed*
                Enqueue q in Q_N
            Else
                $\lambda^+(q) \leftarrow 0$, $\lambda^-(q) \leftarrow 0$
            End If
        End For
        //update previous threshold noise pixel
        For every pixel q in $Q_{N_{\text{check}}}$ for which $N(p) = true$
            If q still belongs to a path of maximum length G in the *deactivated* set
                // upstream and downstream values are not changed
                Enqueue q in Q_N
            Else
                $N(q) \leftarrow false$
                $\lambda^+(q) \leftarrow 0$, $\lambda^-(q) \leftarrow 0$
                Enqueue q in $Q_{N_{propagation}}$
                noise pixel enhancement : $I(q) \leftarrow threshold$
            End If
        End For
        // update $Q_{N_{check}}$ "Ping-Ponging" process
        $Q_{N_{\text{check}}} \leftarrow Q_N$
    End If
    // Propagate path considering $Q_{N_{propagation}}$ pixels as seeds
    For every pixel q in $Q_{N_{propagation}}$
        Rpropagate(q, λ^+, n^+, n^-)
        Rpropagate(q, λ^-, n^-, n^+)
        Q_c pixels check process
    End For
    // Propagate from seed pixel p
    Rpropagate(p, λ^+, n^+, n^-)
    Rpropagate(p, λ^-, n^-, n^+)
    Q_c pixels check process
End For

Figure 4.13: d-dimensional robust path openings

Rpropagate function

function Rpropagate(p, λ^+, n^+, n^-)
// we can both enqueue noise and active pixels in Q_q
enqueue in Q_q all neighbors $p_+ = p + n^+$ for which $b(p_+) = true$ or $N(p_+) = true$
For every pixel q in Q_q
// for each pixel in Q_q *we take the maximum of its downstream neighbors*
$l \leftarrow \bigvee_i \lambda^+(q + n^-(i)) + 1$
If $l < \lambda^+(q)$
$\lambda^+(q) \leftarrow l$
// we can both enqueue noise and active pixels in Q_q *iteratively*
enqueue in Q_q all neighbors $p_+ = p + n^+$ for which $b(p_+) = true$ or $N(p_+) = true$
enqueue q in Q_c
End If
End For
End function

Q_c elements checking process

For every pixels q in Q_c
If $\lambda^+ + \lambda^- - 1 < L$
If $b(q) = true$
// as $G < L$ q *cannot be a noise pixel at a further level*
// as $\lambda^+ + \lambda^- - 1 < L$, q *cannot be a "seed" at a further level*
$I(q) \leftarrow I(p)$
$\lambda^+(q) \leftarrow 0$, $\lambda^-(q) \leftarrow 0$
$b(q) \leftarrow false$, $\overline{s}(q) \leftarrow true$, $\overline{N}(q) \leftarrow true$
End If
If $N(q) = true$
$\lambda^+(q) \leftarrow 0$, $\lambda^-(q) \leftarrow 0$
$N(q) \leftarrow false$, $\overline{s}(q) \leftarrow true$, $\overline{N}(q) \leftarrow true$
noise pixel enhancement : $I(q) \leftarrow I(p)$
End If
End If
End For

Figure 4.14: robust path propagation

5

Selective and Robust Path Operators

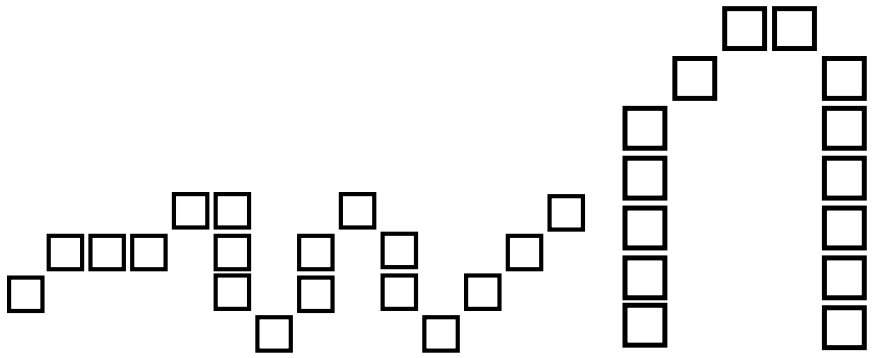
5.1 Motivations

We have shown in the previous chapters that path operators provide a very efficient way to filter thin, elongated and not necessarily perfectly straight features in a n-dimensional image. We have also proposed a method to cope with the main limitation of path operators i.e. path disconnections due to noise that would prevent it from being propagated. However, there still exists some limitations to path operators for both complete and robust versions. This limitation was recently pointed out by Morard *et al.* in (76) and (77).

What make path openings efficient is their ability to be locally adaptable to the geometry of the features while keeping a selectivity in the orientation. Using the union of the response of the complete path operators considering several orientations performs an anisotropic detection of the features of interest.

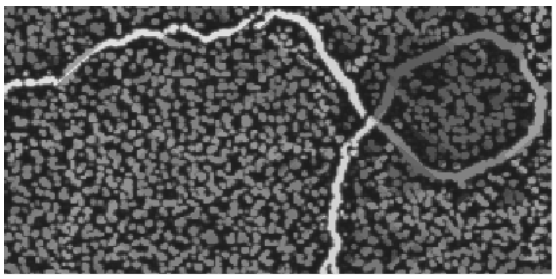
However, for several types of shapes belonging to the so-called "tortuous shapes", both complete and robust path operators will not be able to follow several sudden changes in global features' orientations thus leading to the unintended removal of some features (see Figure 5.2 for an illustration).

Figure 5.1 is an example of two tortuous features. Recently, Morard and al. in (76) and (77) have extended attribute-based operators to geodesic attributes. These attributes are based on the computation of the geodesic diameter (95) for each of the connected



(a) (b)

Figure 5.1: Example of two different "tortuous" shapes, from path openings' point of view, these shapes present several changes in their global orientation and are hard to "follow"



(a)

Figure 5.2: Example of the inability for robust path openings to detect "tortuous" shapes

component of the image.

In this framework, they have also proposed a new geodesic measure called the "geodesic tortuosity" enabling the detection of "tortuous" features by computing the ratio between the geodesic diameter and the euclidian distance between two geodesic extremities. However, the main limitation of these operators is their non-robustness to contrast variations due to noise (in fact the same limitation as complete path operators enunciated earlier in this dissertation).

Considering these recent advances in the domain of connected filters and also those of robust path operators, we propose to combine these two approaches in order to make

use of their respective advantages i.e. the local robustness of path operators and the ability for geodesic-based attribute filters to detect the "tortuous" shapes.
In this chapter, we will first recall the basic notions of attribute-based operators and especially those related to geodesic attributes. Then, we will show how to combine path operators and geodesic attribute-based operators to render the path operators robust to the tortuosity. This new algorithm will be then extended to the greyscale case.
Finally, applications in 2D and 3D will be shown for biomedical imaging.

5.2 Attribute-Based Operators

5.2.1 Connected Components

Let $I : E \to B$ be a binary image, with $E \subseteq \mathbb{Z}^2$ image domain and $B = \{\text{true}, \text{false}\}$. Defining X of I as $X = \{x \in E | I(x) = true\}$, a connected component X_i is defined as the set of elements of E that are *true* and connected according to a given adjacency relation. Generally, in the 2D case, this adjacency relation is defined as either a 4-neighbourhood or 8-neighborhood adjacency relation (in 3D we use a 26-neighbourhood adjacency relation).

5.2.2 Attribute Operators And Thinnings

For a connected component $X_i \in E$ of the foreground, an attribute operator is performing a transform of the image driven by the computation of a binary predicate (C_λ). The criterion checks how an attribute computed on X_i compares to a threshold value (e.g. $C_\lambda := Att(X_i) > \lambda$).
A binary attribute operator Att_λ is defined as follows:

$$Att_\lambda(X_i) = \begin{cases} X_i & \text{if } X_i \text{ satifies the criterion } C_\lambda \\ \emptyset & \text{otherwise} \end{cases} \tag{5.1}$$

Based on this definition, (96) introduced attribute thinnings ρ^{Att_λ}:

$$\rho^{Att_\lambda} = \bigvee_X Att_\lambda(X_i) \tag{5.2}$$

Attribute thinning respects the property of being idempotent and anti-extensive. However, depending on the chosen criterion, the increasingness property is not always respected, otherwise, it would become a so-called attribute opening (e.g. considering the

area which is an increasing criterion leads to an attribute opening). Note that the dual transform of thinning is called thickening and is not studied here, as the results on this operator would be directly inherited from those of attribute thinning by complementation.
To overcome the problem of not being increasing, arbitrary rules can be applied to these operators to extend them to the grey-scale case, (77), (97), (98), (96):

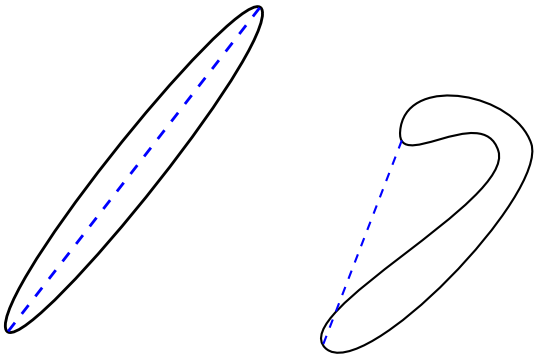
- **Min rule:** A connected component X_i is removed if $Att_\lambda(X_i) = \emptyset$ or if it exists a connected component X_j such that $X_i \in X_j$, which is removed.
- **Max rule:** A connected component X_i is removed if $Att_\lambda(X_i) = \emptyset$ and all the the CCs X_j such that $X_j \in X_i$ are also removed.
- **Direct rule:** A connected component X_i is removed if and only if $Att_\lambda(X_i) = \emptyset$. This is the classical rule used for the grey level decomposition of an opening.
- **Subtractive rule:** A connected component X_i is removed if $Att_\lambda(X_i) = \emptyset$. All the other connected component such that $X_j \in X_i$ are lowered by the value of the contrast of X_i.

It has been shown that the subtractive rule is more convenient in many applications as it provides a better contrast of features of interest. However, in the case of our study, we have chosen to use the direct rule to extend the selective path operators to the grey-scale case as the final goal is to combine both attribute thinnings and path operators that are already use this rule (as usual for an opening).
Note also that the fundamental difference of path operators compare to attribute-based operators is that path operators consider the image at pixel level, i.e. in the same connected component, different pixels can belong to paths of different lengths.

5.2.3 Geodesic Attributes

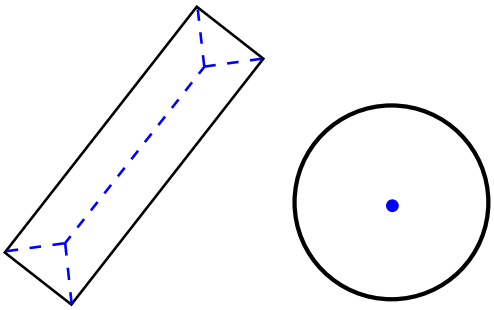
5.2.3.1 Geodesic Distance and Diameter

In this section we will address the problem of answering the question : “How long is an object?”.
Basically, a logical answer should be by computing the length between its extremities. Unfortunately, this method, suitable for simple shape objects can not be generalized to



(a) (b)

Figure 5.3: (a).Simple shape object; (b).Twisted shape object

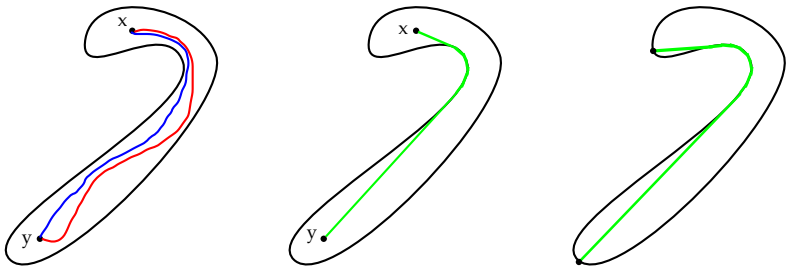


(a) (b)

Figure 5.4: Medial axis of a (a).Rectangle; (b).Circle

tortuous or twisted objects (see Figure 5.3). Another response would be by counting the number of pixels belonging to the medial axis of an object. However, medial axis computation is very sensitive to small shape variations and would lead to an erroneous measure of the length depending on the shape of the object (e.g. the medial axis of a circle is a single point, see Figure 5.4).

Defined by Lantuejoul and Maisonneuve (95), the geodesic distance of a connected component (usually called object) provides a characterization of its length.
Basically, considering two points x and y into an object X, the geodesic distance is the length of the shortest (also called geodesic) arc between x and y.



(a) (b) (c)

Figure 5.5: (a).Two possible paths within the object; (b).Geodesic arc between x and y; (c).Geodesic diameter

For an object, the geodesic diameter $L(X)$ is defined as :

$$L(X) = \sup_{(x,y)\in X} dX(x,y) \tag{5.3}$$

with $dX(x,y)$ corresponding to the geodesic arc between x and y (see Figure 5.5). Compare to the other methods described above, the geodesic diameter has some advantages:

- its definition is general and is applicable to objects of any shape,
- its definition is robust to small changes in the shape of the object,
- its computation can be used to access to other attributes characterizing the object.

5.2.3.2 Geodesic Elongation

The geodesic diameter gives some information about the length of an object, however it is not sufficient to characterize its shape. The geodesic elongation was proposed in (95) and enables the computation of an objective measure of the elongation of an object:

$$E(X) = \frac{\pi L^2(X)}{4S(X)} \tag{5.4}$$

The more elongated the object is, the higher is the geodesic elongation. Note that the geodesic elongation of a disk is minimal and is equal to 1.

5.2.3.3 Geodesic Tortuosity

From the geodesic diameter, (77) and (76) proposed to derive another measure called geodesic tortuosity τ. It is defined as the ratio between the geodesic diameter and the euclidian distance between the two geodesic extremities of the object that minimize this distance. Let $PE(X)$ be the set of all the pair of the geodesic diameter extremities:

$$PE(X) = \{\{x_0, y_0\}, \{x_1, y_1\}, ...\} \tag{5.5}$$

From $PE(X)$ we can define the minimal Euclidean distance :

$$L_{Eucl}(X) = \min_{\{x,y\} \in PE(X)} ||x, y|| \tag{5.6}$$

with $||.||$ corresponding to the classical Euclidean distance in $\mathbb{Z}^2$. The tortuosity $\tau(X)$ is then defined as:

$$\tau(X) = \frac{L(X)}{L_{Eucl}(X)} \tag{5.7}$$

5.2.3.4 Practical Considerations

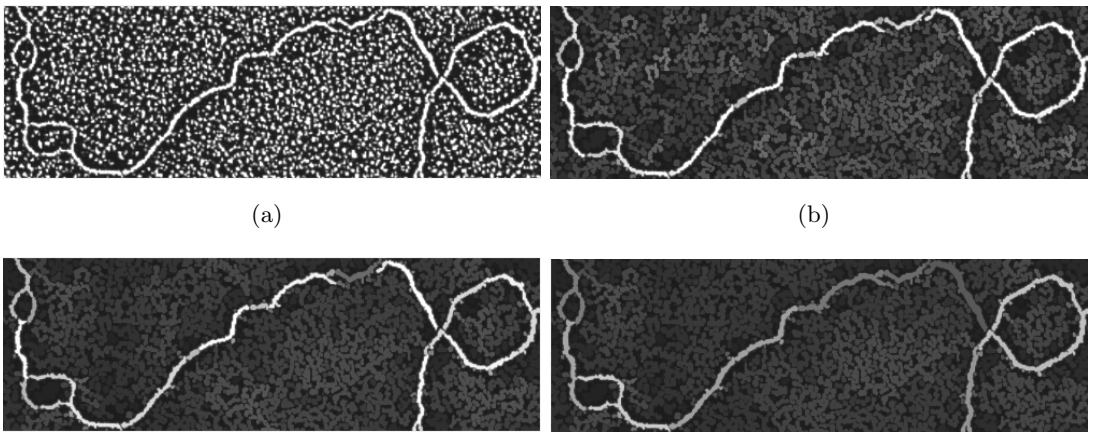
The direct implementation of the geodesic diameter is computationally intensive (see (77) and (76)). Indeed it is equivalent to perform a region growing process considering each of the border pixels in order to build a distance map. Taking the highest level of this distance map would give the geodesic diameter. In order to keep efficiency in the implementation we propose to use an approximation of the geodesic diameter: the barycentric diameter proposed in (77).

5.2.4 The Barycentric Diameter

5.2.4.1 Definition

Recently, (77) proposes an approximation of the geodesic diameter called the barycentric diameter based on an iterative scheme of geodesic distance propagation.
Given an arbitrary location x in a connected component X, let $l_x(X)$ be the maximal geodesic arc from x in X. Considering Y as the set of the elements of X such that $dX(x, y) = l_x(X)$, the maximal iterated geodesic distance is defined as:

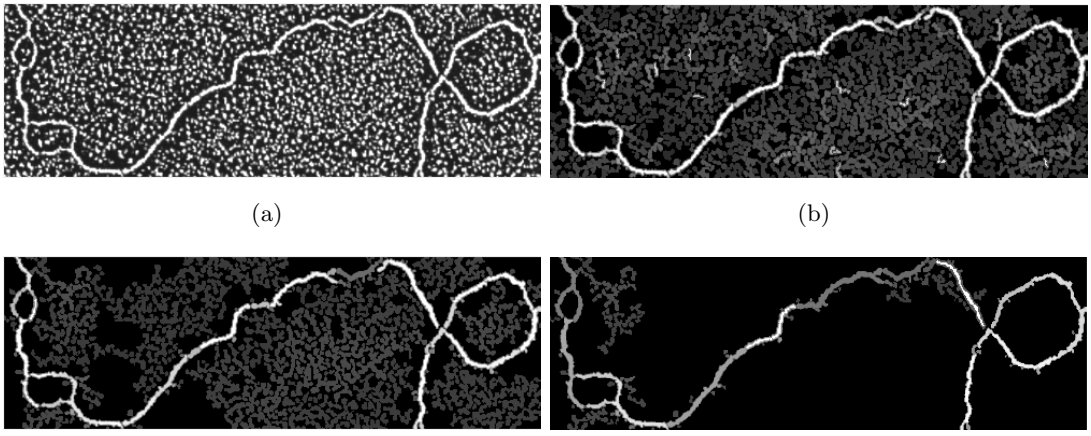
$$l_x^2(X) = \sup_{y \in Y} l_y(X) \tag{5.8}$$



(a) (b)

(c) (d)

Figure 5.6: Result of attribute filtering using the barycentric diameter (a).original image; (b).$L = 20$; (c).$L = 50$; (d).$L = 100$



(a) (b)

(c) (d)

Figure 5.7: Result of attribute filtering using the elongation computed from the barycentric diameter(a).original; (b).$E = 5$; (c).$E = 10$; (d).$E = 20$

The iterative process is converging, not necessarily to $L(X)$, though, in practice, the convergence is fast. Moreover, for practical applications, going beyond $l_x^2(X)$ Combination?is not essential since the barycentric diameter at the second order is already giving a good approximation and will drastically reduce the computation time.

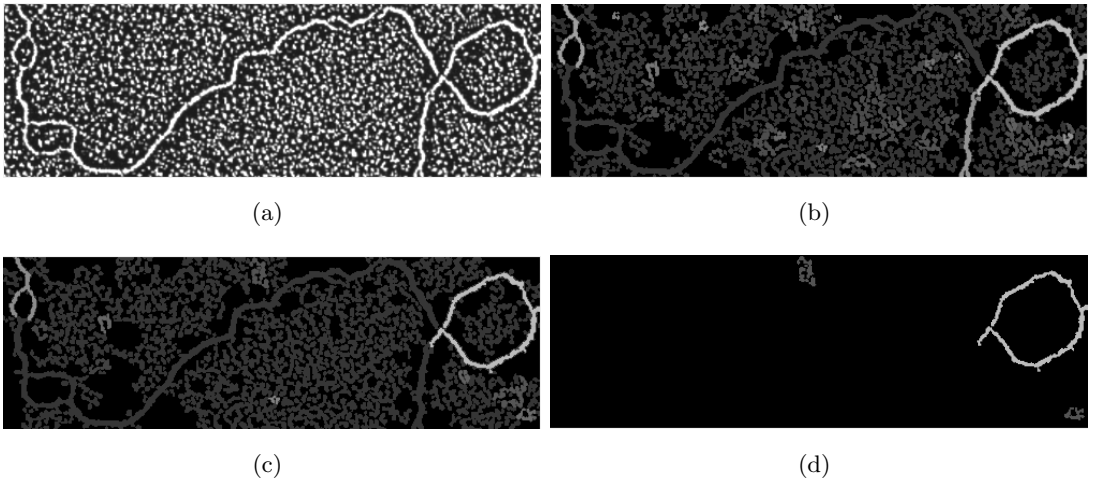


Figure 5.8: Result of attribute filtering using the tortuosity computed from the barycentric diameter(a).original; (b).$\tau = 2.5$; (c).$\tau = 5$; (d).$\tau = 10$

We have given the basic definition of the barycentric diameter starting from an arbitrary location x in X. Several strategies have been tested to find a relevant location to start the iteration process. Considering the barycenter of X, $L_{bar}(X)$ is the barycentric diameter starting at the farthest location from the barycenter. It has been proven experimentally that $L_{bar}(X)$ would provide the best approximation of the geodesic diameter.
This approximation will be used in the implementation of the selective and robust path operators.

5.2.4.2 Geodesic Attributes Derived from the Barycentric Diameter

As described above, we can derive the same attributes from the barycentric diameter than from the geodesic diameter:

$$E(X) = \frac{\pi L_{bar}^2(X)}{4S(X)} \tag{5.9}$$

$$\tau(X) = \frac{L_{bar}(X)}{L_{Eucl}(X)} \tag{5.10}$$

Figures 5.6, 5.7, 5.8 show that attribute filter is able to detect the tortuous objects.

5.3 Selective and Robust Path Operators

5.3.1 Tortuosity and Path-Based Operators

Figure 5.9 illustrates a tortuous shape. In this case, neither complete nor robust path operators are able to cope with several successive changes in shape orientation as it clearly underestimates features' lengths. However, geodesic tortuosity performs well on this type of shapes. The tortuosity in this case equal to 3.5 ($L(X) = 14$ and $Leucl = 4$). Based on this result, it seems natural to use the geodesic tortuosity to combine with

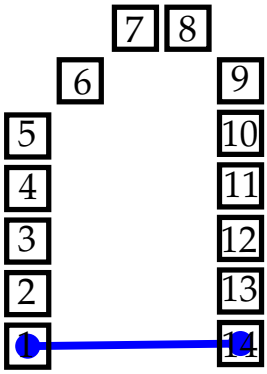


Figure 5.9: Propagation of the geodesic diameter into a "tortuous" shape

path operators in order to create a novel operator thus combining the advantages of the local robustness and features' tortuosity detection.

5.3.2 Principle

Principle behind the selective and robust path operators comes from that of the morphological greyscale reconstruction (99). Following the threshold decomposition principle enunciated previously in this dissertation, L.Vincent defined the greyscale reconstruction as follow:

Definition : Let J and I be two greyscale images defined on the same domain, taking their values in the discrete set $\{0, 1, ..., N-1\}$ and such that $J \leq I$. The greyscale reconstruction $\rho_I(J)$ of I from J is given by:

$$\forall p \in \Re^n, \rho_I(J)(p) = \max_{k \in [0, N-1]} p \in \rho_{X_k(I)} X_k(J) \tag{5.11}$$

with $\rho_{X_k(I)}X_k(J)$ the binary reconstruction operator at threshold k obtained by iterating, until stability, an elementary geodesic dilation of the marker image into the mask image:

$$\rho_{X_k(I)}X_k(J) = \bigcup_{n\geq 1} \delta^{(n)}_{X_k(I)}X_k(J) \tag{5.12}$$

Selective and robust path operators aim at combining robust path-based filtering with a morphological reconstruction provided by a pre-computed tortuosity map for each of the connected components (see Figure 5.10 for an illustration on a 1D signal).

By integrating a geodesic reconstruction in the robust path operators algorithm, we can make use of both active and noise pixels connectivity to retrieve features that respect a particular attribute. Note that here we will focus our study on the use of a tortuosity map but it could be the geodesic diameter, the elongation or the original image itself.

The key point of this algorithm is in the definition of the geodesic reconstruction by using the notion of threshold superposition. This definition will help us to design an efficient d-dimensional algorithm for greyscale images in order to integrate the reconstruction algorithm by using the threshold decomposition performed inside the robust path operators algorithm.

5.3.3 Definition of the Greyscale Selective and Robust Path Operators

Here we propose the theoretical foundations of the selective and robust path-based operators. We recall that we can define the set of G-robust paths of length L in X as:

$$\alpha^G_L(X)_{\text{Robust}} = \bigvee\{\sigma(\boldsymbol{a}) \cap X : \boldsymbol{a} \in \Pi^G_L(X)_{\text{Robust}}\} \tag{5.13}$$

With a slight modification, we can take into account the connectivity through the noise pixels set in E:

$$\alpha^G_L(X)_{\text{Robust}\,E} = \bigvee\{\sigma(\boldsymbol{a}) : \boldsymbol{a} \in \Pi^G_L(X)_{\text{Robust}}\} \tag{5.14}$$

Greyscale Robust and Selective Path Operators:

The greyscale and robust selective path operators is defined as being the result of the morphological greyscale reconstruction considering the result of the robust path

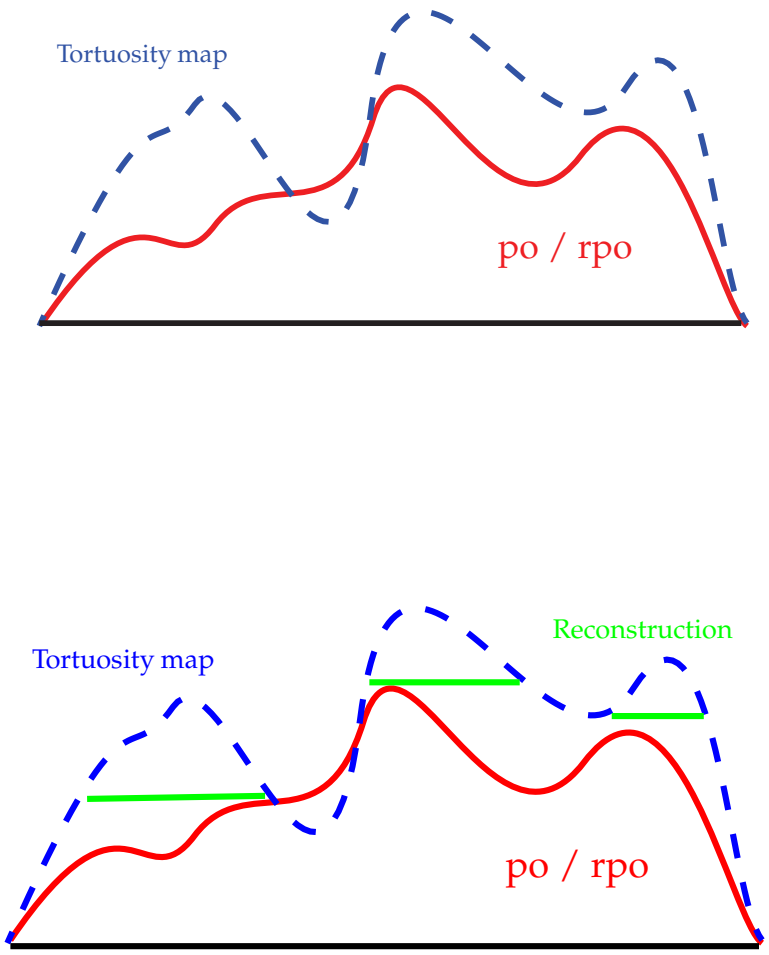


Figure 5.10: Principle of the Selective and Robust Path Operators

operators as marker and the *supremum* of the result of the robust path operators and the tortuosity map as mask (see Figure 5.10 for an illustration).

$$\boldsymbol{\alpha}_L^T(X) = \rho_{(\alpha_L^G(X)_{\text{Robust}\,E} \vee T)} \alpha_L^G(X)_{\text{Robust}\,E} \tag{5.15}$$

As the result of an opening by reconstruction, this operator obviously respects the

canonical properties of an opening.

5.3.4 Non-Increasing Attribute Operators

Figure 5.11 illustrates the principle used in the conception of the grey-scale selective path operators. The input image is represented as a stack of connected components, the vertical and horizontal axis represents the grey level value and the connectivity respectively. Each block on a line is a connected component at a given grey level value. For both attribute and path-based operators, elements (connected components in the case of attribute filter and pixels in the case of path operators) are highlighted (blue and green for attribute and path-based operators respectively). Here we have decided to remain consistent with the strategy used for an opening to extend the binary selective path operators to the grey-scale case i.e. by using the direct rule. Using this strategy for the attribute image, it is equivalent to say that if a pixel status is *true* at a given grey level value $t1$, it will be *true* for all grey levels $t2$ respecting $t2 \leq t1$. We will use this assumption to build the grey-scale selective path operators.

5.3.5 Algorithm

In this section we describe the algorithm for the greyscale selective and robust path operators. Algorithm 5.12 uses the threshold decomposition procedure performed in the implementation of the robust path operators to operate a morphological geodesic reconstruction of the result of the robust path operators into the image given as mask. At a given threshold T, once the update procedure of the robust path operators is over (i.e. the deactivation of the pixels that do not respect the criterion of being part of a robust path of parameters (L, G)), a geodesic reconstruction is performed using a FIFO queue propagation from the $\{active + noise\}$ set left at threshold T into the set of mask pixels whose value $\geq T$. This algorithm is similar to what can be found in the state of art (see e.g. for a complete description of these algorithms (99)).
Note that the fundamental difference between this algorithm and a typical geodesic reconstruction by dilation is that we can make use of the connectivity trough noise pixels to recover the destroyed part of the features (see Figure 5.13 for an illustration).

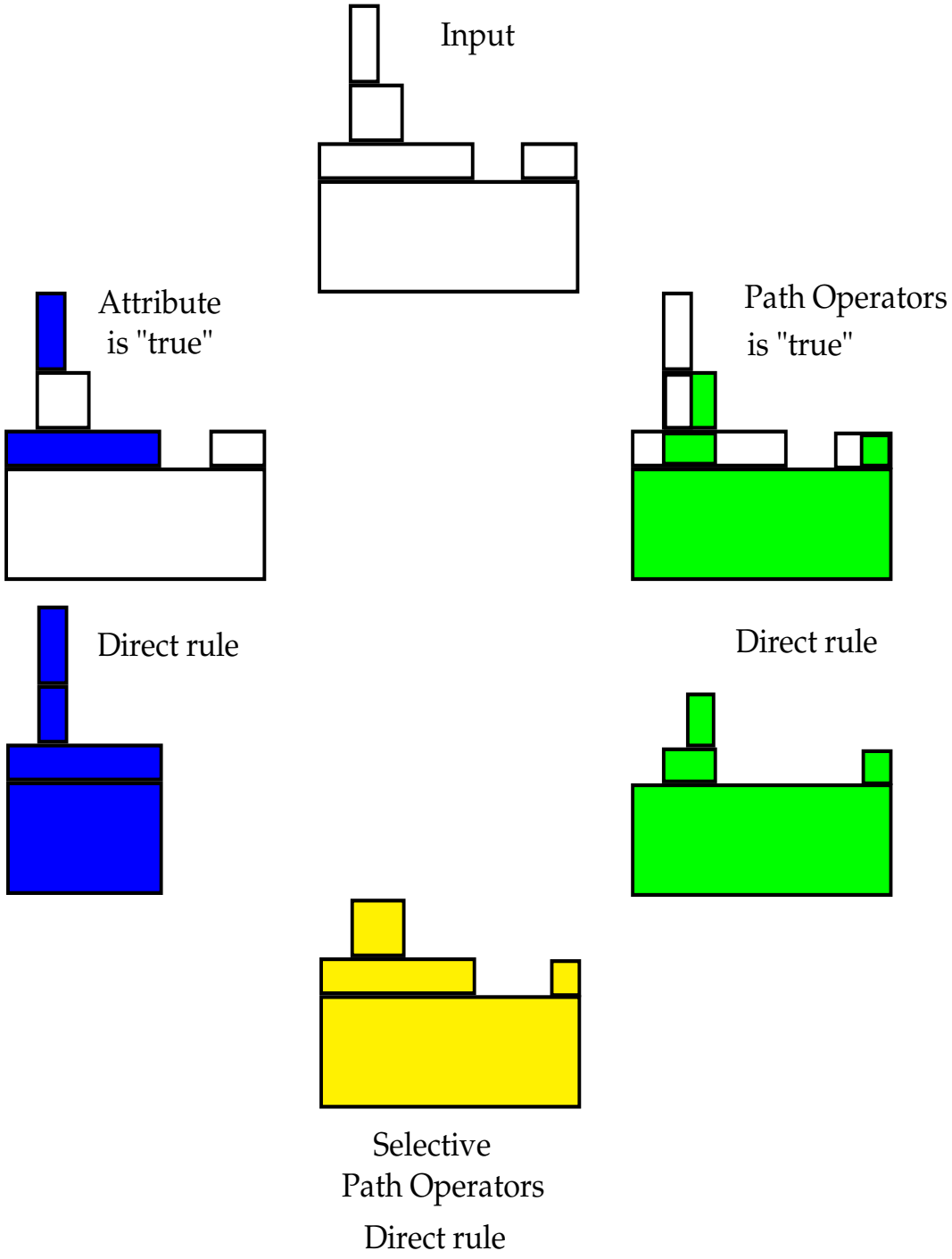


Figure 5.11: Grey-scale selective path operators

list of parameters:
I is the input greyscale image
$Mask$ is a greyscale image (e.g. tortuosity map)
T is the current threshold in the threshold decomposition of RPO
base-idx is the current pixel index in the threshold decomposition of RPO
idx are sorted in ascending order (with respect to image values)
N_G contains a list of neighborhood indexes for a given grid G (2D or 3D)
Q_{prop} fifo propagation queue storing pixels indexes
$flagP$: boolean image (1 if already propagated, else 0)
function Reconstruction( )
for $idx = base\text{-}idx$, $idx < image\text{-}size$
  if $b(idx) == true$
    Q_{prop}add() $\leftarrow idx$, $flagP(idx) \leftarrow$ true
    while Q_{prop} not *empty*
      $idx\text{-}p \leftarrow Q_{prop}$first()
      for $q \in N_G$
      if $(Mask(idx\text{-}p + q) \geq T \;||\; b(idx\text{-}p + q) \;||\; N(idx\text{-}p + q))$ and $\overline{flagP(idx\text{-}p + q)}$
        Q_{prop}add() $\leftarrow idx\text{-}p + q$, $flagP(idx\text{-}p + q) \leftarrow$ true
        if $\overline{b(idx\text{-}p + q)}$ $\overline{N(idx\text{-}p + q)}$ $\{I(idx\text{-}p + q) = T\}$
    Q_{prop}pop()
end for
reset $flagP$

Figure 5.12: Greyscale Selective and Robust Path Openings : Propagation Function

5.3.6 Results and Timings in 2D and 3D for Biomedical Imaging

In this section we present some examples of applications of the robust and selective path operators for 2D and 3D biomedical imaging.

The first example (see Figure 5.14) illustrates its use on a 2D retinal image for vessel detection. Based on this result it is clear that the path operators underestimate tortuous features' lengths (due to their intrinsic nature of being formed by a constrained connectivity) whereas geodesic tortuosity performs well on this type of shape (note that elongated features are not detected). The geodesic tortuosity gives a very strong response to the tortuous features of the image, however, elongated features are completely discarded. Selective and robust path operators succeed well in detecting

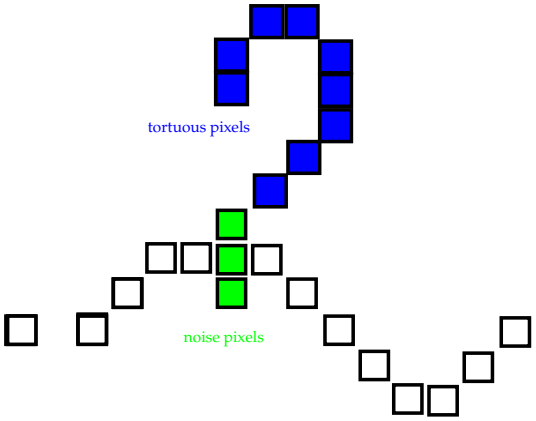
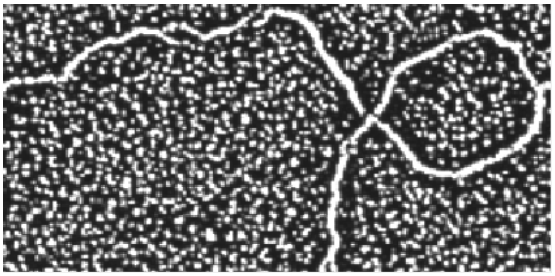


Figure 5.13: Selective and Robust Path Operators extend the connectivity trough noise pixels

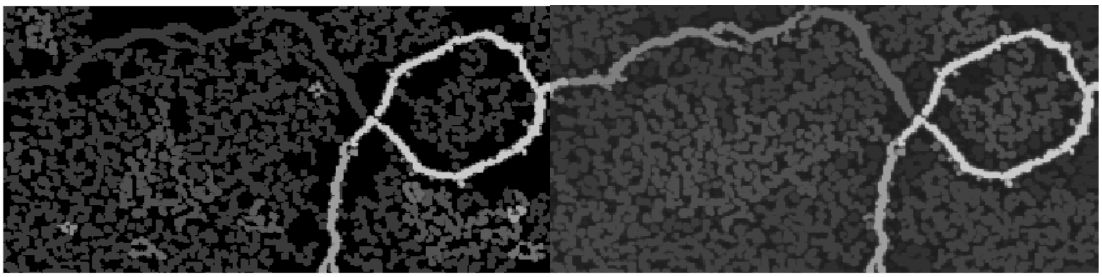
elongated, noisy and tortuous features. Note that on this image (size 500×160×8-bit) the union on four orientations is performed in less than one second.
Another example of their efficiency is given for the extraction of vessels in a 3D angiogram image (see Figure 5.15). Timings for one orientation on a 256×256×256×8-bit is 9.5s.

5.3.7 Discussion

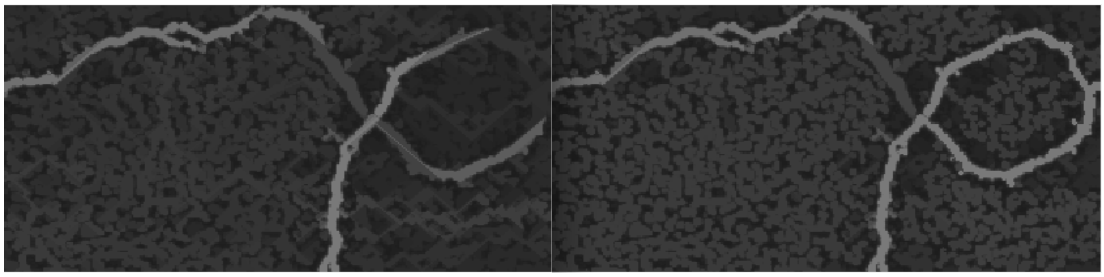
In this chapter we have proposed an original contribution, the selective and robust path operators aiming at combining attribute and robust path based operators to cope with the limitations for path-based operators not being robust to "tortuous" features. Moreover, we benefit of the connectivity through noise pixels of the background to propagate the geodesic reconstruction which render this operators very powerful and more flexible than a traditional geodesic reconstruction by dilation (e.g. it could be useful to use it on the output of a Hessian based filter who generally fail in recovering the junctions for the detection of tortuous brain vessels). Given timings and results on both 2D and 3D images have proven their usefulness in practical applications of biomedical imaging.



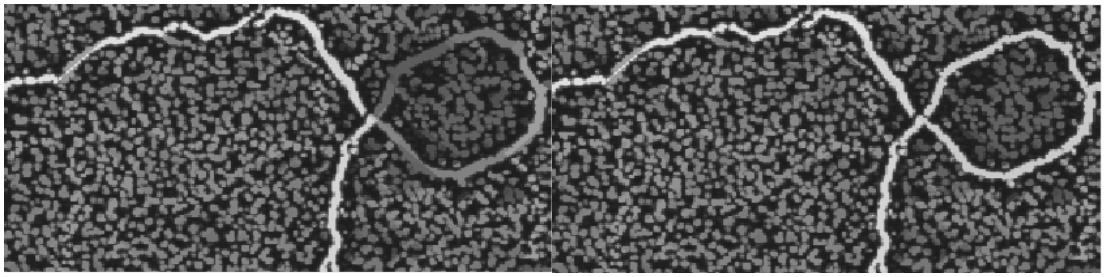
(a)



(b) (c)

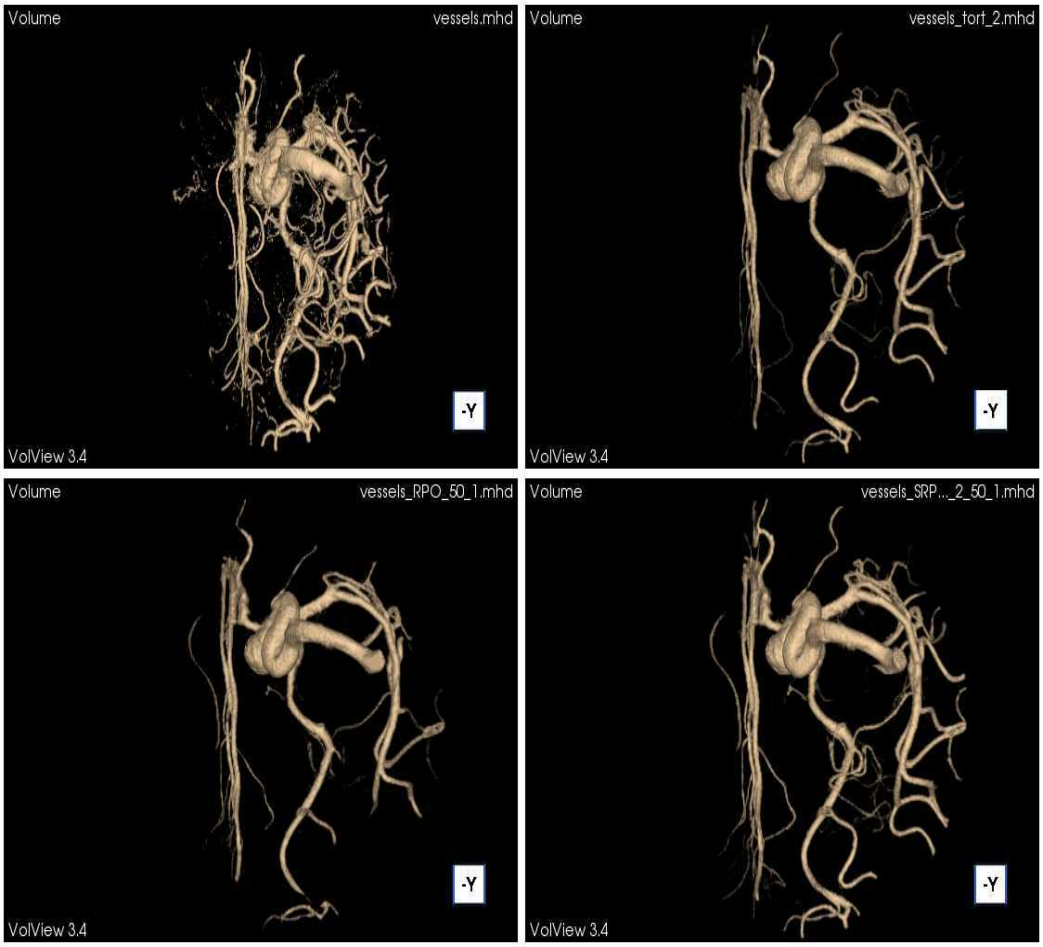


(d) (e)



(f) (g)

Figure 5.14: Results and comparisons; (a).Input image; (b).Tortuosity ($\tau = 3$); (c).Barycentric diameter ($L = 150$); (d).Complete path operators ($L = 100$); (e).Selective path operators ($L = 100$ and $\tau = 3$); (f).Robust path operators ($L = 150$, $G = 2$); (g).Selective and robust path operators ($L = 150$, $G = 2$ and $\tau = 3$).



(a)

Figure 5.15: Results and comparisons in 3D; From left to right and top to bottom: Surface rendering of : original image, tortuosity attribute ($\tau = 2$), RPO ($(L, G) = (50, 1)$) and SRPO.

6

Applications

6.1 The DELPIX Project

X-Ray tomography has been widely used in medical imaging for decades as it was clearly a revolution in this domain. So far this technique was not very spread in the industry. This was mainly due to the large amount of additional time needed to perform the reconstruction of the volume from the sequence of projections, the low resolution and the cost.
In this industrial context, the goal of the DELPIX project (formed by two public research laboratories and specialized companies in the field of image processing and tomographic reconstruction) is to bring this new technology to industrial partners by providing breakthroughs in most of the technological limitations.
Our research comes at the end of the project chain. Providing efficient algorithms for 3D image processing will enable industrial partners in material science to process 3D samples in order to perform automated analysis.

6.2 Characterization of Three-dimensional Fibrous Material

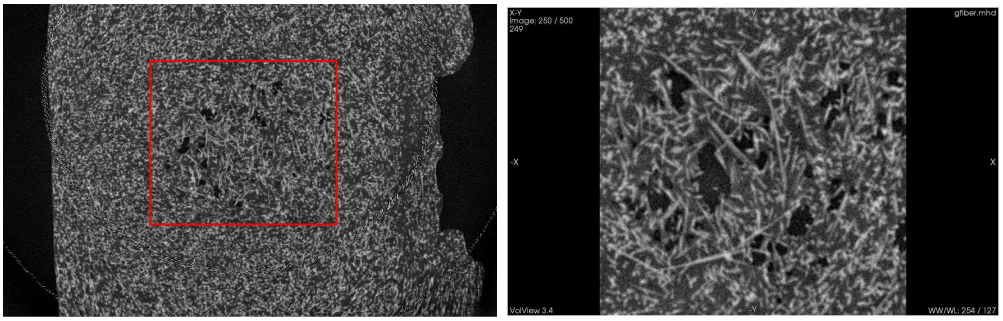
6.2.1 Framework of the Study

With recent breakthroughs in micro and nanotomography, practitioners in material science can have an access to precise information about material microstructure. More than non-destructive testing, these information can also lead to the design of material

with improved properties.
Due to the huge amount of data generated by 3D computed tomography, reliable and automatic analysis is essential for the practitioner in material science.
Here we will propose a method to extract the characteristics of three-dimensional fibrous material on a real case. This task is still challenging as we can encounter a huge complexity and randomness into fibers' network microstructure.

6.2.2 The Material

Glass fibers reinforced polymers are widely used in aerospace and automotive industry. The main challenge is to provide a very high rate of mechanical performance compare to production cost.
However, the main difficulty when producing this material is to ensure that glass fibers are correctly spread into the polymer matrix. Their good repartition at the microscopic scale will induce a good transfer of mechanical stresses at the macroscopic scale. They also have an important impact on the thermal property of the material. Fibers orientations, lengths and distribution over space are then considered as critical.
X-Ray computed tomography is a non-destructive imaging process enabling the representation of the glass fibers and the polymer matrix at a micrometer scale. A specimen is placed on a rotary plate between the X-Ray source and the detector. An image is then acquired by using the 2D projection of the specimen at a particular angle (angle is increased step by step). At the end of the acquisition step, each projection is used to generate a grey-scale 3D image which corresponds to the spatial X-ray attenuation coefficient of the specimen.
In the case of our study, the material is widely used in industrial applications and is made of polyamide reinforced with 30% of glass fibers. Initially the raw sample data is $1900 \times 1200 \times 2000 \times$ 8-bit, the acquisition was performed with a resolution of $3.5\mu m$. To reduce the computation time, a volume of interest of $500 \times 500 \times 300$ voxels corresponding to the central part of the raw volume (red square on Figure 6.1(a)) was extracted.
Figure 6.1(b) represents a slice of the extracted volume.
As one can notice, the contrast between the fibers and the polymer matrix is very low and their orientations seem randomness. Moreover their characteristic size is near of the limit of the resolution of the acquisition system (about 7 voxels for the diameter



(a) (b)

Figure 6.1: Raw data under study: (a).Slice of the whole volume; (b).Slice of the extracted volume;

and 110 voxels for the length which represents 24μm and 420μm respectively).
However we can note that fibers are straight features and their shapes and intensities are locally altered by the noise mostly produced by the acquisition device and the reconstruction process. As the density of fibers in the volume is relatively high, they are heavily entangled thus resulting in a lot of touching areas which makes harder the extraction of their individual characteristic.

6.2.3 Existing Methods for Fibers' Characteristics Extraction

Extracting fibers characteristics in 3D is a well-treated problem in the literature.
Axelsson, (100), (101), estimates paper sheets' fibers orientations by using a local measure of orientation with quadrature filters and structure tensors. This method is applied directly on the grey-scale image and make the hypothesis of small signal variations in the direction of the fiber. In the same idea (102), (103) propose to compute the local orientation in each voxel by convolving the grey-scale image with anisotropic Gaussian kernels thus enabling to have access to both fibers' shapes and orientations distributions.
In (65) Soille and Talbot are using the union of oriented segments to find local structure orientations in a 2D image. The computation of the response for a sequence of rotated filter gives an indication on the local orientation. Moreover, varying the length of the structuring element involved enables to have an indication on the length of the features. In 3D, this approach may result in a computationally unworkable algorithm due to the

increase of the amount of data to treat and to the increase of orientations one would have to consider to reach an acceptable result.
Mulat *et al.* (104), (105) proposed an algorithm for the detection of the 3D axis of tubular features relying on the gradient and curvature estimation. This method has been successfully applied to the extraction of the geometrical properties of carbon-fiber reinforced carbon composites. However it supposes a strong coaxial gradient.
Glass fibers reinforced polymers have also been studied. Most of existing methods (106), (107) rely on the analysis of the skeleton of the segmented fibers. These methods have been proven to be efficient to give a good estimation of fibers' characteristics, however working directly on the skeleton reveals to be very delicate.
Going farther with this idea, (108) and (109) are using a graph based modeling of the skeleton to extract fibers characteristics individually. This method has been proven to be efficient for synthetic images.

6.2.4 Proposed Method

Due to the nature of the material used in this study: low contrasted image, low gradient, fibers at the limit of the resolution of the imaging system, noisy environment; methods based on the local estimations of fibers orientations are not well adapted especially at touching areas. We have decided to use a method based on the simplification of a skeleton model of the segmented volume of fibers thus combining (108) and (106), (107).
The preprocessing step is critical in this method as it aims at enhancing the contrast between the fibers and the polymer matrix. Classical edge-preserving smoothing and background removal steps will be followed by robust path-based operators. After segmentation, a skeletonization transform is applied and is used to perform a simplification of the fibers volume while preserving its topological properties. A modeling step based on graph is then performed on the skeleton and provide a convenient way to filter out noise issuing from the skeletonization process and then to simplify the representation.
A visualization program using the VTK (Visualization Toolkit) library (110) was also implemented in order to display the 3D graph model obtained at different steps of the process.

6.2.5 Preprocessing Pipeline

As we have seen in FIgure 6.1, X-Ray computed tomography generates a large amount of noise during the acquisition and the reconstruction process. To simplify the segmentation step and to increase *in fine* the accuracy of the modeling performed on the glass fibers, we provide a preprocessing pipeline based on morphological path-based filters. Figure 6.2 gives an overview of the preprocessing pipeline. More details for each step are given below.

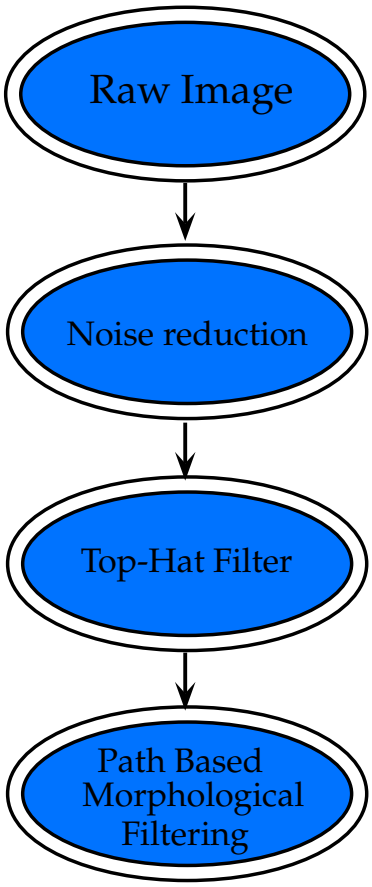


Figure 6.2: Overview of the pre-processing diagram

6.2.5.1 Noise Reduction

To smooth the image while preserving the edges, we make use of the bilateral filter with parameters $\sigma_d = 4.0$, $\sigma_r = 20$ according to the geometric and radiometric aspects of the content of the image. For the sample data, the bilateral filter took approximately

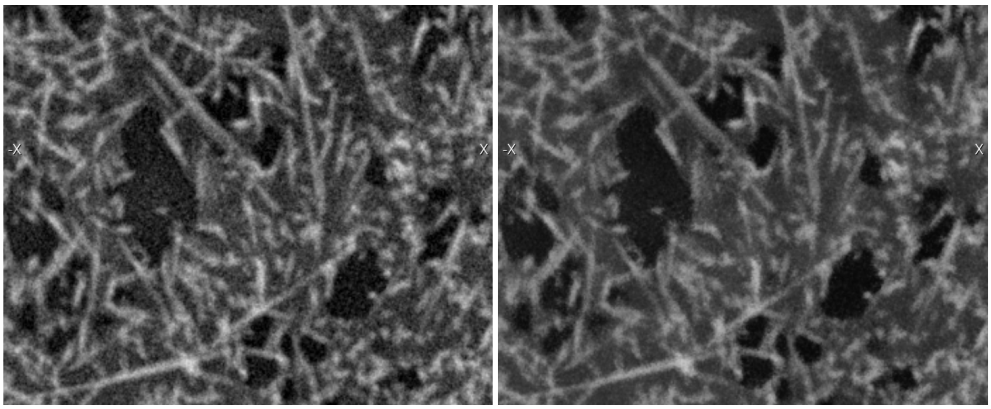


Figure 6.3: Result of the bilateral smoothing; Left: Raw image slice; Right: Result of the smoothing performed by a bilateral filter;

5 minutes.

6.2.5.2 Top-Hat Filter : Extraction of the Local Maxima

An isotropic white top-hat WTH is applied on the image to detect the local maxima which are smaller than the defined structuring element. This step copes with the non-uniformity of the background.
Considering a spherical structuring element b and a set of pixels X, the WTH is computed as :

$$WTH_b(X) = X - \gamma_b(X) \tag{6.1}$$

We have chosen a spherical structuring element of radius equal to 4. With this parameter we ensure that almost all the fibers are detected. However, as the polymer matrix is obviously not perfectly uniform, the WTH image reveals to be noisy. Indeed, as we haven't integrated any strong constraints on the shape, each of the local maxima suppressed by the opening will be retrieve in the WHT image (see Figure 6.4).
After this operation, we have enhanced the contrast between the matrix phase and the

fiber phase, however fibers result to be locally low contrasted (disconnected). For our image the top-hat filter took approximately 3 minutes.

6.2.5.3 Path-Based Morphological Filtering

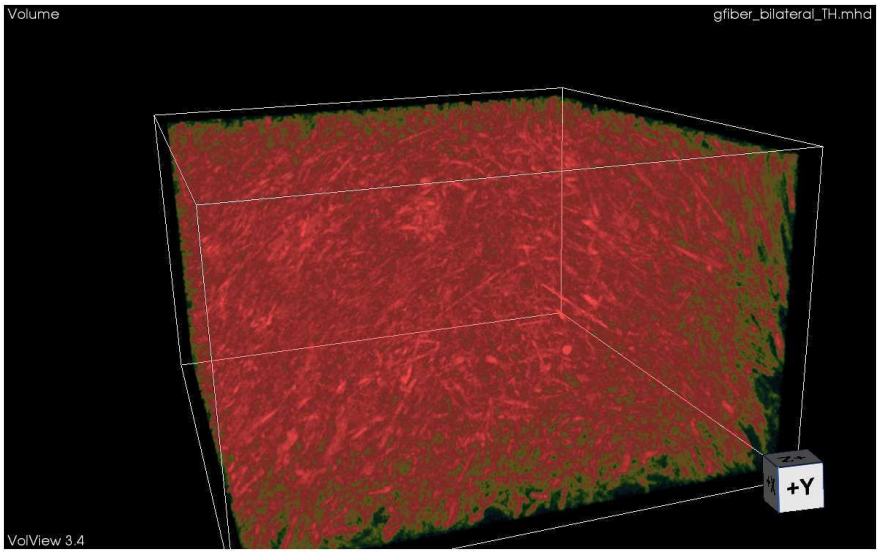
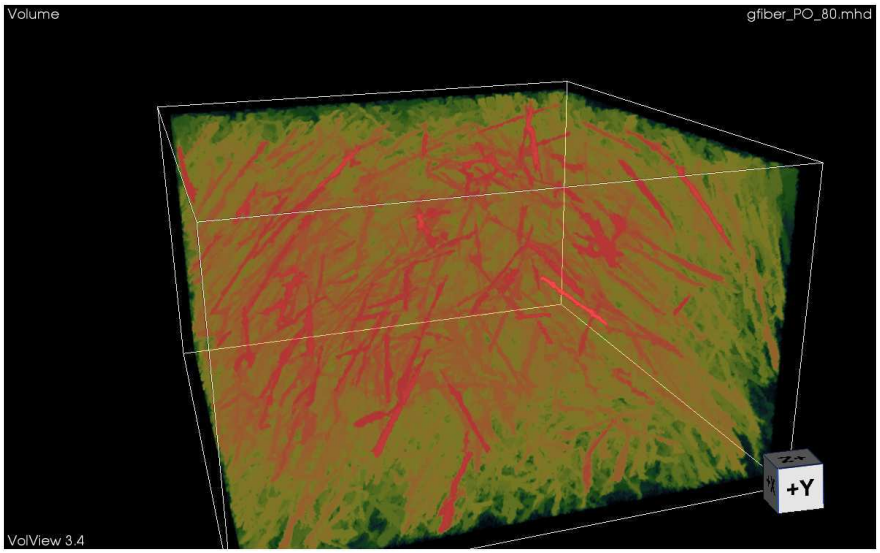
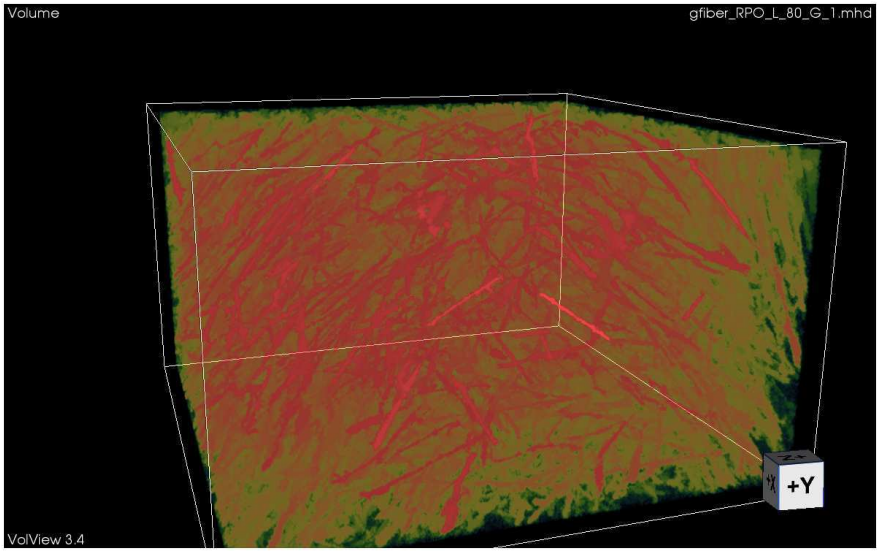


Figure 6.4: Result of the WTH operator

To reduce the noise issuing from the WTH image, we have to integrate the notion of shape when filtering the image. Fibers are linear-like features which are locally disconnected by noise. The use of a morphological filter able to filter the noise while keeping intact features of interest should be very efficient and would provide a good input image for the segmentation step.
While 3D rank-max openings may provide a satisfactory result, they would fail in detecting the locally non-straight parts of the fibers. Moreover their use would induce the discretization of the 3D space depending on the number of filters involved which reveals to be a delicate step.
As we have no *a priori* on fibers' orientations, we have decided to use 3D robust path operators to retrieve the fibers while removing a maximum of compact noise.
Practically, we compute the union of robust path operators considering 7 different orientations (3 corresponding to the main axes of the images and 4 to the main diagonals) with the parameter $L = 80$ and $G = 1$. These parameters are chosen to provide a



(a)



(b)

Figure 6.5: Path-based operators applied on the WTH image; (a).Complete path operators ($L = 80$); (b).Robust path operators ($L = 80$ and $G = 1$); Note that the same color mapping was used.

sufficient radiometric flexibility while not increasing the detection of correlated noise too much (see Figure 6.5). We have also processed the top-hat image with complete

path operators considering the same detection length and the same orientations.
For a $500 \times 500 \times 300$ 8 bits grey-scale image, the algorithm took approximately 29 and 14 minutes for the robust and complete path operators respectively. Note that in both cases we are computing the union on 7 different orientations considering the same path length.

6.2.6 Segmentation

The preprocessing pipeline has reinforced the contrast between the fibers and the polymer matrix and has removed a large amount of noisy features. Here we use an hysteresis thresholding (HT) to segment the image.
Hysteresis thresholding is an advanced thresholding algorithm using two thresholds and an adjacency relation between pixels. We believe that this segmentation is well adapted to our case of study as the output of the preprocessing contains fibers with local disconnections.
Given $\tau 1$ and $\tau 2$, the output value of a pixel x is defined as :

$$HT(I(x)) = \begin{cases} 1 & \text{if } I(x) \geq \tau 2 \\ 1 & \text{if } I(x) \geq \tau 1 \text{ and } \exists \text{n-path}(x,y) | I(y) \geq \tau 2 \text{ and } \forall p \in \text{n-path}(x,y),\ I(p) \geq \tau 1 \\ 0 & otherwise \end{cases} \tag{6.2}$$

We have manually set the parameters to $\tau 1 = 55$ and $\tau 2 = 70$ to ensure that the lower contrasted part of the fibers were retrieved, we use the 26-neighborhood adjacency relation as usual in the 3D case.

6.2.7 Basic Concepts of Discrete Topology and Skeletonization

Before going on the description of our method, we will give here basic concepts of discrete topology needed for the understanding of what follows. Even if we have used in previous parts of this dissertation some of the concepts given below, we need here a more formal definition. More details about discrete topology can be found in (111).
Considering A a point of a discrete space $E = \mathbb{Z}^d$ with $d = 2, 3$. We can define the classical adjacency (or neighborhood) relations in 2D and 3D by $N_4(A)$, $N_8(A)$ and $N_6(A)$, $N_{18}(A)$, $N_{26}(A)$ respectively (see Figure 6.6).
Further in this dissertation we will denote as α a number such that $\alpha = \{4, 8, 6, 18, 26\}$. We define $N^*_\alpha(A) = N_\alpha(A) \setminus A$ and B is said to be α-adjacent to A if $B \in N^*_\alpha(A)$. A

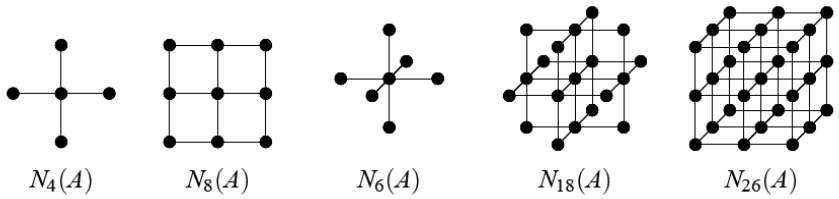


Figure 6.6: Adjacency relations in 2D and 3D

α-path is a sequence of points $A_0...A_k$ such that A_i is α-adjacent to A_{i-1}. Stating that $X \subseteq E$, A and B of X are said to be α-connected in X if there exists an α-path in X between these two points. Note that this relation is a symmetric but not reflexive binary relation. A subset X of E is said to be α-connected if it is composed by one α-connected component. The set of all α-connected components is defined as $C_\alpha(X)$ and a subset $Y \subseteq E$ is said to be α-adjacent to $A \in E$ if there exists a point $B \in Y$ such that B is α-adjacent to A. The set of α-connected α-adjacent components to A is noted $C_\alpha^A(X)$.

The concept central to the skeletonization transform is the concept of *simple points* (112), (113), (114). A point is simple if its removal does not change the number of connected components of X and of its complementary. This property can basically be determined by a local characterization of the point i.e the number of connexity $T_\alpha(A, X)$ and $T_{\overline{\alpha}}(A, \overline{X})$. Indeed, a point A is α-simple if $T_\alpha(A, X) = 1$ and $T_{\overline{\alpha}}(A, \overline{X}) = 1$. Roughly speaking it means that A is simple if it has only one component in the "object" set and only one component in the "background". Note that even if the concept of *simple point* is global, it can be characterized locally (this will lead to efficient implementation of digital homotopic transforms as we will see below). Note also that different adjacency relation are used for the object and for its complementary (α and $\overline{\alpha}$) mainly to avoid topological paradoxes of objects being connected and not connected at the same time (111) (usually the 4-adjacency is associated with the 8-adjacency in 2D and the 6-adjacency is associated with the 26-adjacency in 3D).

Going back to our study, we were stopped at the segmentation step. To simplify the representation of the fibers network while keeping its topological properties, we propose to use a skeletonization transform. In this study we will use a sequential thinning algorithm. The basic concept of this family of algorithm is to detect at each iteration

the simple points set and remove them from the object one after the other until no simple points remain. The set of points obtained at the end of the transform is called skeleton as it represents the raw shape of the object.

This basic strategy guarantee the transform to be homotopic however it may not provide a good centering. Indeed the order of simple points removal is critical for sequential thinning. Improved strategies for the removal of simple points have been developed in order to remove them "layer by layer" from the contour to the center of the object. Using a *priority function* based on a distance map (115), (116), (117) is widely used: at each iteration, the simple point which has to be removed is that of lowest priority. Here we have used the implementation provided by the Pink software (Pink software homepage) using an approximate of the Euclidian distance for the priority function (see Algorithm 1).

Algorithm 1 Sequential thinning with priority function

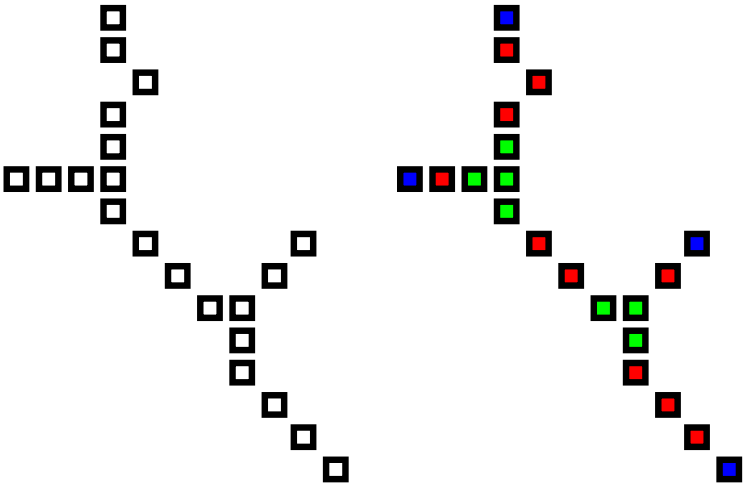
```
1: procedure SeqThinningPrior(X ⊆ E , P : X → ℝ or ℤ)
2:     repeat
3:         Detect A ∈ X such that A is simple for X and that P(A) is minimal
4:         X = X \ A
5:     until Stability
6:     return X
7: end procedure
```

6.2.8 Skeleton Analysis

Skeletonization will reduce the amount of data while preserving the topology of the objects. As the analysis of a raw skeleton from real data is not a fair problem, working on a skeleton model is an easier way to extract fibers individually and to obtain their characteristics.

The first step is to decompose the skeleton into three types of points according to their number of non-zero neighbors giving by the adjacency relation (e.g. 26-neighbourhoood in 3D) : end-points, curve-points and cluster-points have respectively one, two and more than two non-zero neighbors.

Practically, the skeleton is considered as a binary image. An array is created and will associate to each of the pixel a flag to its type. Once the classification is made, we



(a) (b)

Figure 6.7: (a).Example of a 2D skeleton; (b).Result of point classification

are able to aggregate cluster and edge points according to their connectivity and to create the so-called clusters, edges and end-point sets. One can see in Figure 6.7, the classification of end-points, curve-points and cluster-points which are respectively of blue, red and green colors.
After having classified skeleton points, one can notice that the skeleton is made of 'path' and 'clusters'. A 'path' is a connected string of curve-points lying between two 'clusters' which are themselves composed by one point or by an aggregation of points. Basically 'paths' can be classified into three types (see Figure 6.8(a).):

- 'branch-branch' path (B-B) connecting two clusters
- 'leaf-branch' path (L-B) connecting a cluster to an end-point
- 'leaf-leaf' path (L-L) connecting two free clusters (representing an isolated fiber)

6.2.9 Graph Modeling

Based on the work of (108), we have decided to model the fibers skeleton by a graph. A graph G is a pair (V, E), with V and E the set of vertices and edges respectively. Here we model the skeleton by an undirected simple graph. Undirected means that

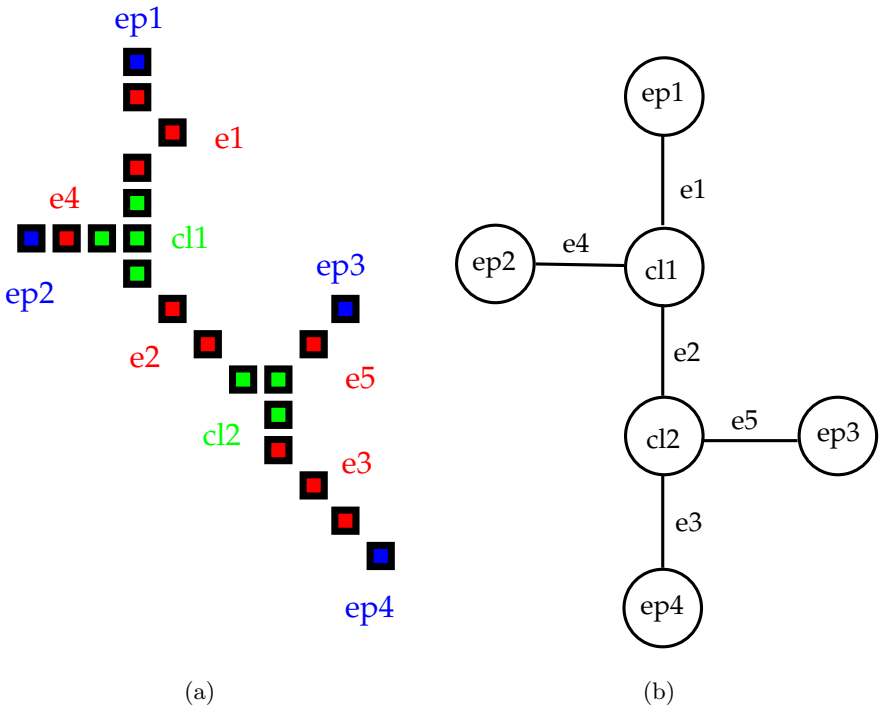


Figure 6.8: (a).Skeleton after points classification and aggregation; (b).Its corresponding graph representation

edges are not oriented (given two vertices $v1$ and $v2$: $(v1, v2) = (v2, v1)$) and we refer as simple, a graph which is undirected and has no loops and no more than one edge between any two different vertices.

In order to be physically coherent with the geometry of the fibers, we associate some properties to vertices and edges. Vertices are associated clusters or end-point positions (for a cluster its position corresponds to its barycenter) $b_0 : V \rightarrow \mathbb{R}^3$. Edges are associated barycentre's coordinates $b_0 : E \rightarrow \mathbb{R}^3$, length $l : E \rightarrow \mathbb{R}^+$, and orientation vector $v_{orientation} : E \rightarrow \mathbb{R}^3$.

The orientation corresponds to the normalized vector between the vertices connected to one edge. We define the degree of a vertex $d(v_i) = \text{Card}\{v_j \in E, j \neq i\}$ as the number of vertices linked to it by an edge. The graph is created using the *Boost Graph Library* (118) which provides template classes for graph instantiation.

A 3D rendering of the graph model can be seen Figure 6.9. One can notice the very complex structure of fibers network as well as distortions affecting the medial axis. We

will dedicate the following part of this chapter to propose some methods to filter the graph while trying to respect the topology of the fibers.

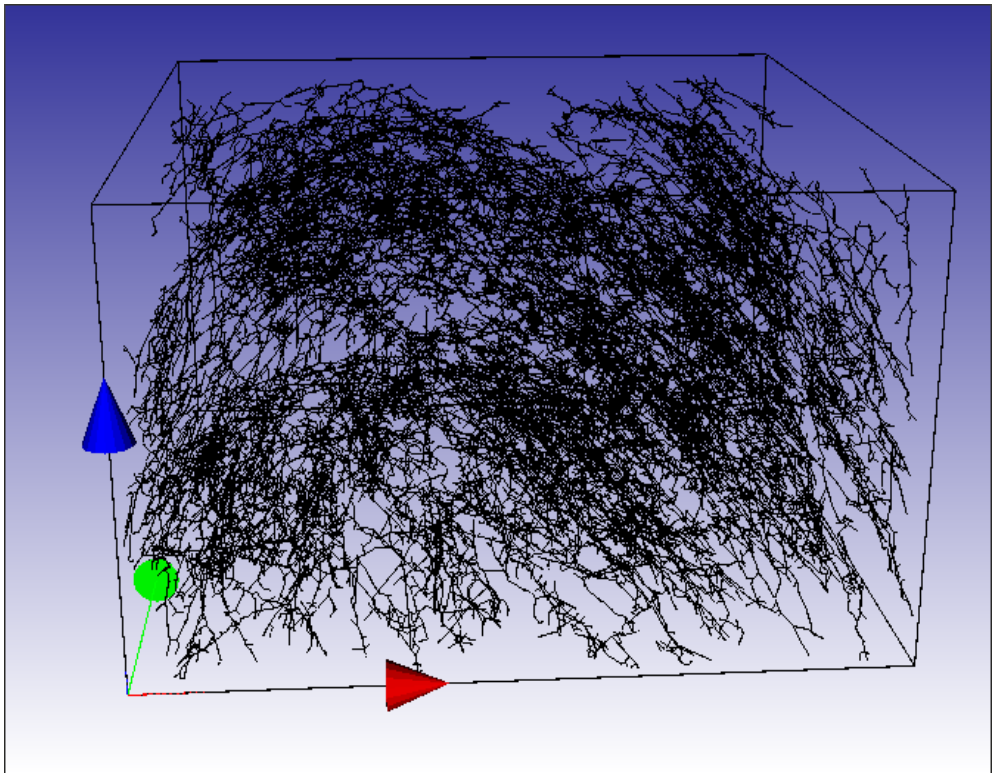


Figure 6.9: 3D rendering of the graph model

6.2.10 Graph Filtering

As skeletonization process generates a large amount of spurious branches (see Figure 6.9), filtering the graph is a necessary step to increase the effectiveness of the fiber extraction algorithm.

6.2.10.1 Surface Irregularities Corrections

During the skeletonization step, fibers surfaces irregularities are generating several spurious branches (see Figure 6.10). This phenomenon classically result in so-called *T-junctions*. *T-junctions* are represented in the graph as a L-B path with a cluster vertex

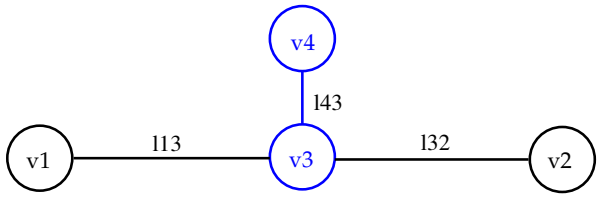


Figure 6.10: Surface irregularity representation

of degree three. After *T-junctions* of length inferior to L_{min} have been identified, they are removed from the graph G following Algorithm 2.

Algorithm 2 Filter out graph edges resulting from surface irregularities

procedure FILTERINGSPURIOUSBRANCHES($graph, L_{min}$)
  for each $e_{i,j} \in E$ **do**
    if ($d(v_i) = 1$ and $d(v_j) = 3$) and $l(e_{i,j}) < L_{min}$) **then** ▷ If we have found a *T-junction*.
      $E \leftarrow E \setminus \{e_{i,j}\}$ ▷ Edge contraction
      $V \leftarrow V \setminus \{v_i\}$▷ Here we consider that v_i corresponds to the single point.
      Compute $e_{k,l}$ parameters from $e_{j,k}$ and $e_{j,l}$
      $E \leftarrow E \setminus \{e_{j,k}\}$ *and* $E \leftarrow E \setminus \{e_{j,l}\}$
      $E \leftarrow E \bigcup \{e_{l,k}\}$
    end if
  end for
  return *graph*
end procedure

6.2.10.2 "Ladder-shape" Pattern Correction

When fibers are crossing at a very acute angle, a so-called "ladder-shape" pattern could be generated (see Figure 6.11). This is represented as a B-B path (the clusters at the extremities are of degree three). Once these edges are identified as belonging to a "ladder-shape" pattern, we just have to remove them from the graph.
The identification is made by analyzing the number of possible matching pair of edges at the cluster vertex according to the angles formed through itself (see Algorithm 3). According to the fact that the fibers are approximately straight, we arbitrary state that this angle (we refer to this angle as "kink-angle") couldn't exceed 30 degree.

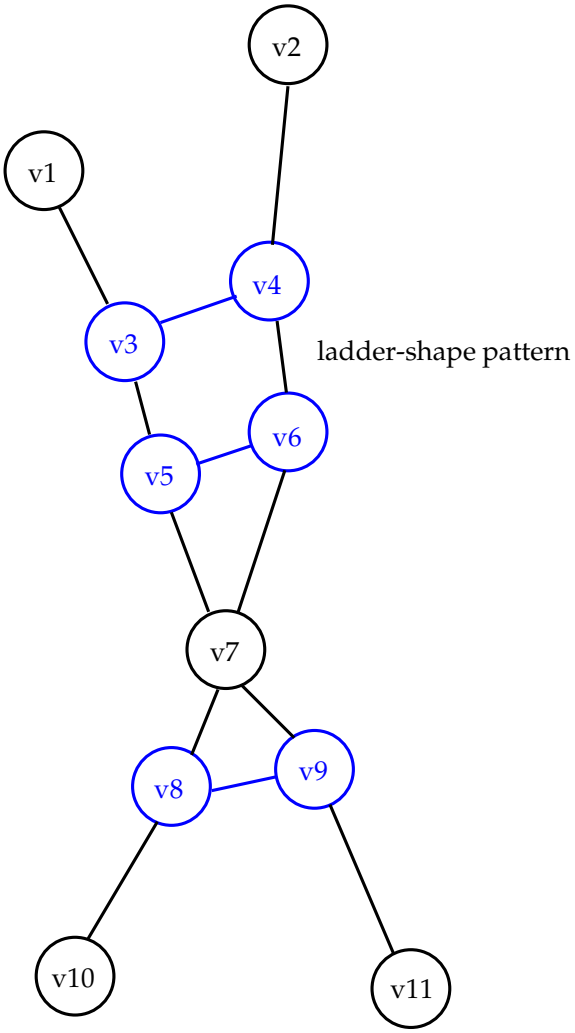


Figure 6.11: "Ladder-shape" pattern representation

6.2.10.3 Crossing Fibers

As fibers have a finite thickness, the skeletonization process produces a false representation of the touching area (see Figure 6.12). In the vicinity of the contact region, the crossing of two fibers is likely to produce two vertices of degree three instead of one cluster of degree four (see Figure 6.12 (b)). To cope with this limitation we propose an edge contraction algorithm capable to correct this erroneous representation while keeping the topology of the fibers. The idea is to simplify the pattern generated by the crossing area. The crossing will be represented as an ideal single vertex in which all

Algorithm 3 Filter out edges belonging to "ladder-shape" patterns

```
1: procedure FilterLadderShapeEdges(graph, kink angle, L_min)
2:     for each e_{i,j} ∈ E do
3:         if (d(v_i) = 3 and d(v_j) = 3) and l(e_{i,j}) < L_min) then
4:             Computation of edges' angle at v_i for out(v_i) \ e_{i,j}
5:             Computation of edges' angle at v_j for out(v_j) \ e_{i,j}
6:             if Angles at v_i < kink angle and v_j < kink angle then
7:                 E ← E \ {e_{i,j}}
8:                 Contract out(v_i) and out(v_j) edges
9:                 V ← V \ {v_i} and V ← V \ {v_j}
10:            end if
11:        end if
12:    end for
13:    return graph
14: end procedure
```

the edges of the crossing will be connected.

Firstly, we create a new vertex ($v7$ in Figure 6.12 (c)) which is the barycenter of the extremities of the edge to contract (here $v3$ and $v4$). Secondly, we remove the connections to the extremities of the edge to contract, and then connect them to the vertex recently created. Thirdly, the lengths and orientation are updated according to the position of the new vertex $v7$. Algorithm 4 is an overview of the process of the simplification of edges representing a false crossing area.

6.2.10.4 Clusters Simplification

The graph G was filtered in order to remove the spurious edges mostly generated from noise and from erroneous crossing surface representation. At this step, the fibers are still entangled.

To simplify fibers network we need to pair fibers segments at each of vertices formed by clusters. Here we are using the fact that glass fibers injected in the polymer matrix are made from rigid material. Fibers will be formed by line segments attached at clusters and almost aligned.

As in the "ladder-shape" filter, we are using a "kink angle" of value 30 degree. This angle is used as a tolerance in the deviation of the fiber axis at cluster point and so,

Algorithm 4 Simplification of false crossing area representation

procedure SIMPLIFYCROSSING($graph, L_{min}$)
  for each $e_{i,j} \in E$ **do**
    if ($d(v_i) = 3$ and $d(v_j) = 3$) and $l(e_{i,j}) \leq L_{min}$) **then**
      $V \leftarrow V \bigcup \{v_k\}$ ▷ v_k is the barycenter of (v_i, v_j)
      $E \leftarrow E \setminus \{e_{i,j}\}$
      Connect all adjacents edges of v_i and v_j to v_k
      $E \leftarrow E \setminus \{$ *adjacent edges of* v_i *and* $v_j\}$
      $V \leftarrow V \setminus \{v_i, v_j\}$
      Update the parameters of the adjacent edges of v_k
    end if
  end for
  return *graph*
end procedure

under this angle, two edges issuing from a cluster have the possibility to be paired otherwise they are considered to belong to different fibers.
Practically, each of the vertices whose degree is strictly superior 1 is considered. A boolean flag *vertice simplification* is initialized to *true* for each of the vertices during graph generation. Cluster simplification algorithm will end when all vertices are simplified i.e. when we can not paired two adjacents edges, or, when no more edges are linked to the considered vertice (in this case the vertex is removed from the graph).

6.2.10.5 Post-processing Filtering

As a result of the previous filtering steps, all the fibers have been detached to each other. In the last step, we will remove the smallest fibers resulting from the clusters simplification algorithm. These edges are mostly due to noise and should not be taken into account in the results.
Practically edges are removed if their length is under a user specified threshold.

6.2.11 Result of the Modeling

Figure 6.14 shows a 3D rendering of the graph model after postprocessing. One can notice that fibers lengths are underestimated regarding the length we can measure on the raw 3D image manually. Main errors generated from this method are coming from

Algorithm 5 Clusters simplification

procedure ClusterSimplifications($graph, kink_angle$)
  for each $v_i \in E$ with $d(v_i > 1$ and $simplification = true$ **do**
    Find $e1$ and $e2$ minimising the deviation through v_i
    if No edges found **then**
      $simplification = false$
    else
      $E \leftarrow E \setminus \{e_{1,i}\}$ and $E \leftarrow E \setminus \{e_{2,i}\}$
      $E \leftarrow E \bigcup e_{1,2}$
      Update parameters for $e_{1,2}$
      if $d(v_i) = 0$ **then**
        $V \leftarrow V \setminus v_i$
      end if
    end if
  end for
  return *graph*
end procedure

the fact that the sample volume used in the study was small compare to the size of the fibers thus don't enable their total representation. Moreover, due to the noisy nature of the input skeleton, simplifications at clusters can lead to the fragmentation of the fibers. Figure 6.13 shows the obtained distribution of both azimuth and elevation angles in the fibers volume. We can observe that angle distributions seem to be in accordance with what we can observe and seem relevant.

6.2.12 Discussion

Even if we have now access to the all the parameters for each of the fibers, we are aware that this method has to be validated. Unfortunately no access is provided to ground truth data as it seems very complicated to generate it manually.

Computationally speaking graph generation and filtering steps are negligible compare to the preprocessing pipeline, segmentation and skeletonization (about 2 minutes against almost 1 hour).

However we have two options that will be studied for future work. Firstly, the use of synthetic data with known characteristics, the goal is then to attain the same measured

values compare to generated ones. This evaluation is quite efficient but synthetic images have difficulties to model all the distortions affecting a real 3D acquisition. Moreover, industrial partners are often more interested in the computation of the parameters on a real case than on a synthetic image. Secondly, we could use the correlation of the results with results obtained with other methods. In our case, we could compare how fits the distribution of orientations estimations to that of methods that are computing local fibers orientations on the input grey-level image. This could give a sufficient level of confidence to validate other fibers' characteristics.

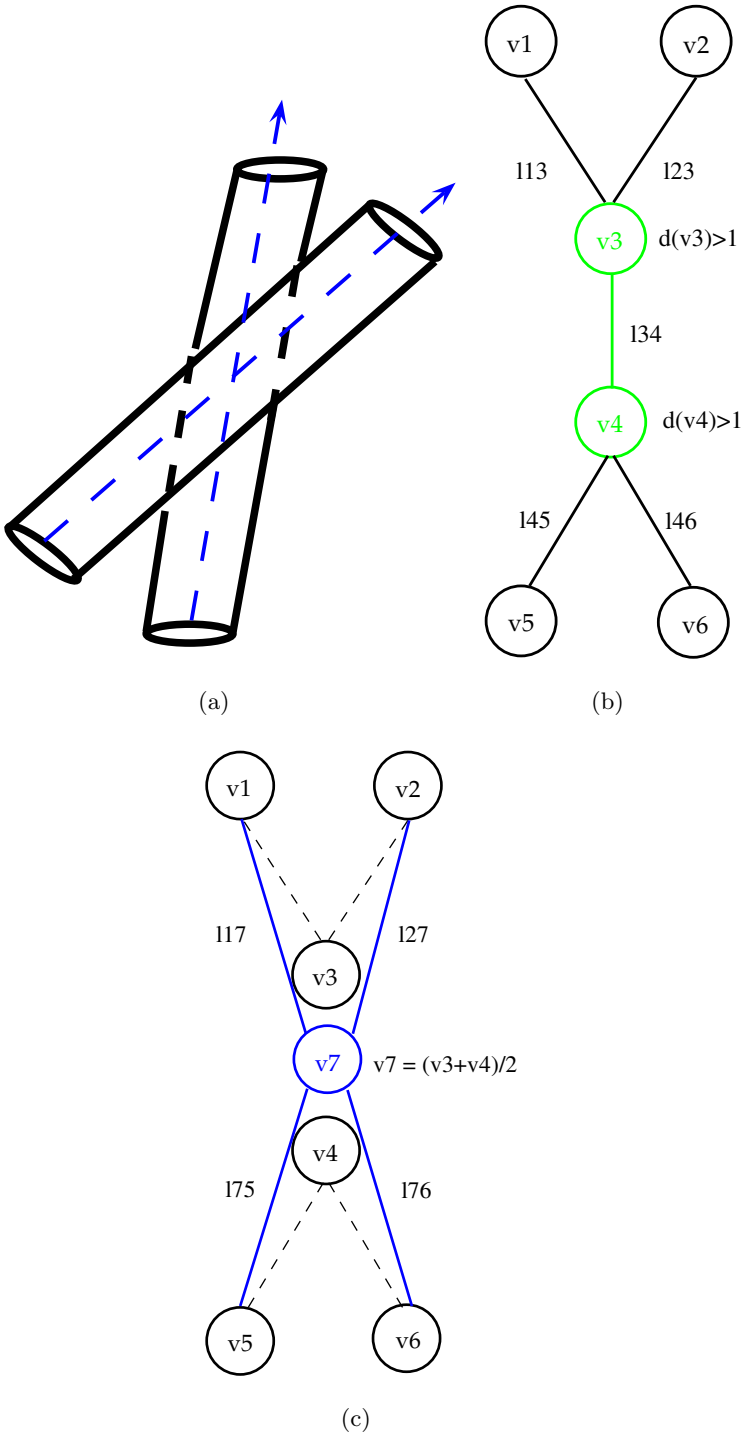
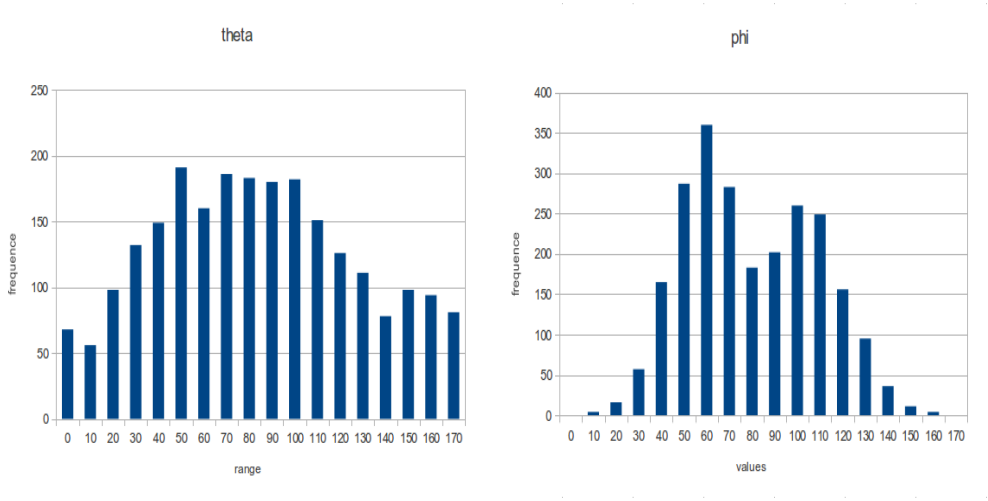


Figure 6.12: (a).3D representation of two crossing fibres; (b). The corresponding graph representation of a crossing after the skeletonization : the green edge corresponds to the false representation of the touching area between the two fibres; (c).Result of the crossing area representation simplification : edge $v7$ is created, $v34$ is deleted and all the edges previously connected to $v3$ and $v4$ are then connected to $v7$.



(a) (b)

Figure 6.13: Azimuth (theta) and elevation (phi) angle distribution

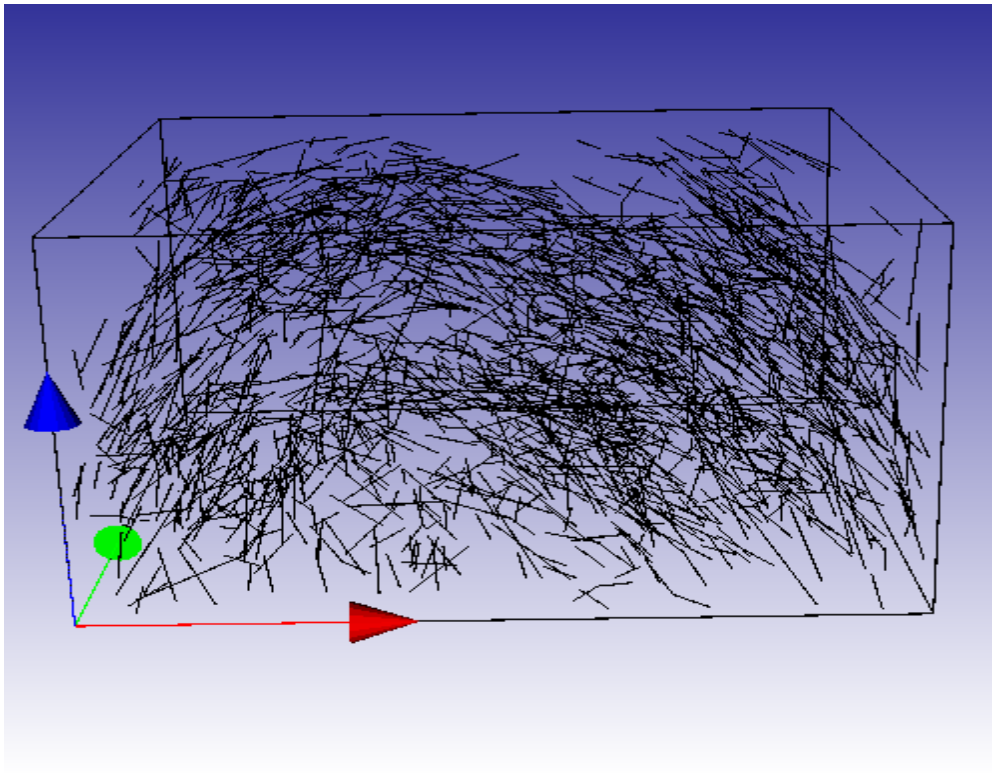


Figure 6.14: 3D rendering of the graph after postprocessing

7

Conclusion and Future Directions

The main purpose of this thesis was to provide some advances in the filtering for 3D image processing applications. Due to the amount of data generated by the acquisition systems and the wish for practitioners in image processing to reach more and more accurate results, there is a real need to provide tractable implementation of complex algorithms.

We have oriented our study on morphological filters which have been proven to be very efficient in medical imaging or material science.

With the benefits of recent advances in 3D morphological filtering in the domain of the detection of thin, curvilinear and oriented features, our contribution in this domain is multiple. We have provided a new framework to cope with disconnections corrupting thin, oriented and flexible noisy features in d-dimensional images by proposing a novel algorithm: the robust path openings and closings. After giving its principle, we have given the theoretical framework of this new operator together with an efficient implementation able to compute d-dimensional images. This algorithm has been shown to generate a larger family of structuring elements than state of art's operators. Their efficiency was shown on both synthetic and real 2D and 3D data. Moreover we have experimentally proven quantitatively their efficiency on 2D retinal images compared to multiscale derivative filters and classical morphological filters.

We have also proposed an extension to robust path openings which can be used to reconstruct the noisy gaps between feature elements. In the case of a robust path openings, the result of this reconstruction operator forms a morphological closing.

At the cost of an increase in computation time, considering recent advances in the field

of connected filters, we have combined these filters with robust path operators to build an operator capable to combine the advantages of both of these approaches. We have proposed to use geodesic attribute-based filters to cope with the main limitation of path operators of not being capable to detect "tortuous" features. The result operator is the selective and robust path operators.

As an application in the field of non-destructive testing, we have integrated robust path openings into an image processing pipeline for the characterization of glass fibers reinforced polymer. They have shown to be well adapted to the enhancement of fiber which is a critical issue for further steps as segmentation, modeling and analysis.

A lot of care in this work has been given in order to develop computationally efficient implementation.

Even if we have provided efficient implementations of new morphological filters, there is a need for practitioners in 3D image processing to run fast implementations of advanced methods. A good way to improve drastically the speed of the algorithms can be found in their parallelization. Future work on these algorithms would be to find a way to generalize the parallel programming of propagation based algorithms such as path operators or geodesic diameter.

In this dissertation we have made a separation between 'detecting' and 'preserving' filters in order to classify them. However, it could be very interesting to combine different approaches coming from the scale-space theory methods and morphological methods. In (71), morphological oriented segments and a vesselness measure are combined in order to provide a very efficient 'morpho-Hessian' filter. The use of the robust path operators in place of oriented segments could maybe be more convenient in vessels detection applications.

Appendix A

Résumé en Français

A.1 Introduction

Avec l'émergence des nouveaux systèmes d'imagerie, nous sommes maintenant capables de générer des images 3 dimensions (3D) qui nous permettent d'observer des phénomènes se déroulant de l'échelle macroscopique à l'échelle nanoscopique.
Dans un premier temps conçu pour l'imagerie médicale, la tomographie par rayons X est devenue fortement répandue comme moyen très efficace et peu invasif de diagnostiquer certaines pathologies (des tumeurs par exemple).
Ce système d'imagerie a récemment trouvé des applications industrielles dans le domaine de l'automobile ou encore de l'aéronautique où la prédiction des comportements mécaniques de certains matériaux est devenu critique.
La question qui reste en suspens est maintenant de savoir quelles informations pouvons-nous extraire de ces volumes de données 3D et avec quels outils.
En deux dimensions (2D), l'analyse et le traitement des images a déjà prouvé son efficacité notamment dans les domaines de l'imagerie médicale ou encore de la télédétection.
L'opération de filtrage en traitement d'images est probablement une des opérations les plus importantes. Cela consiste à supprimer le 'bruit' pour ne garder qu'un maximum d'information utile dans l'image. Bien évidemment, cela est complètement dépendant de l'application considérée, les outils utilisés doivent donc être adaptés á chaque cas d'étude.
Au prix d'une augmentation du temps de calcul et d'occupation mémoire, les implémentations des opérateurs classiques en 2D ont leurs équivalents en 3D. On peut penser par exem-

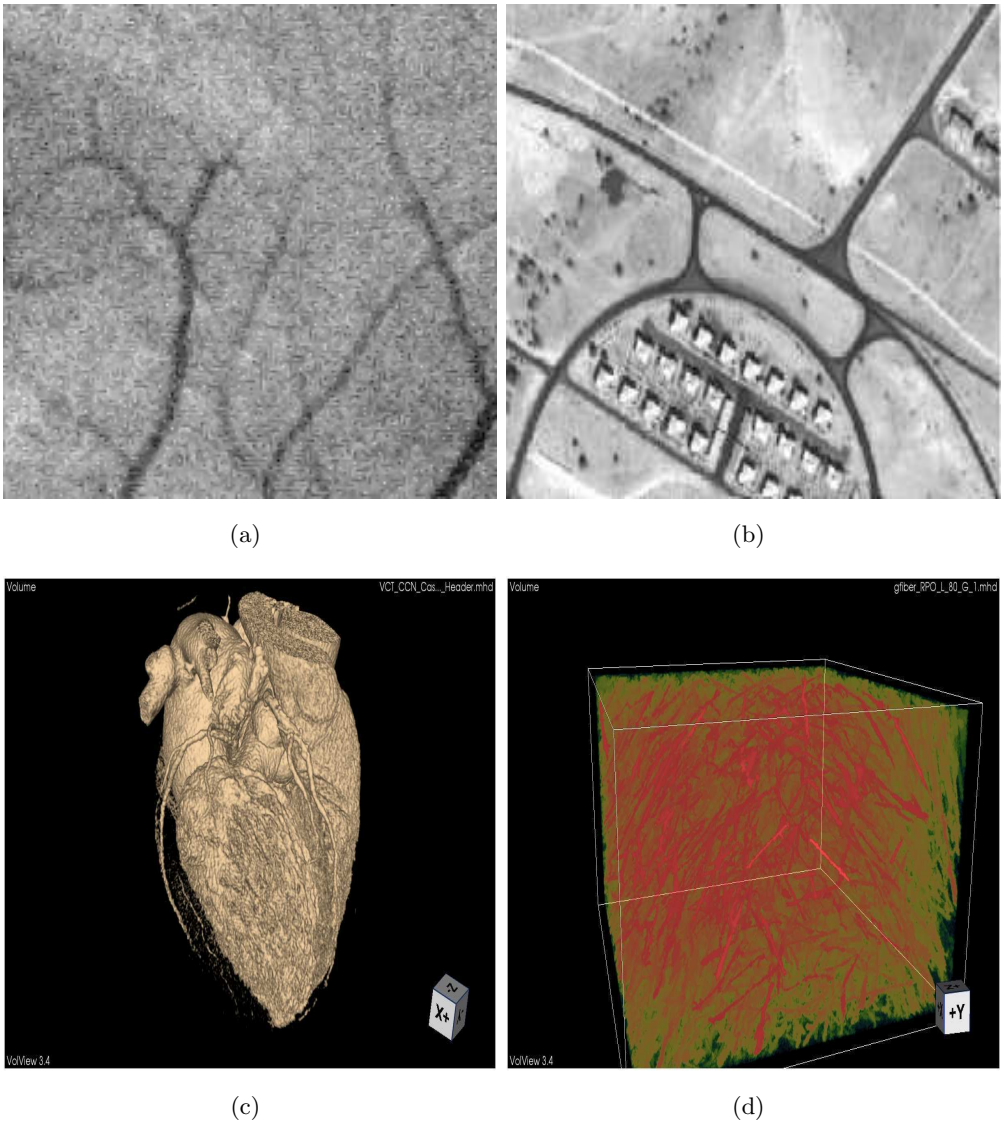
ple aux opérateurs dérivatifs de Canny et Deriche.
Cependant, pour accroître la fiabilité des mesures effectuées dans des cas de plus en plus complexes, des méthodes avancées doivent être utilisées qui souvent, aboutissent à des implémentations inéfficaces en 3D.
Pour faire face à cette difficulté, des implémentations parallélisées ont été développées, malheureusement cela ne concerne qu'une certaine classe d'algorithmes.
Dans les travaux de recherche qui vont suivre, nous allons proposer plusieurs nouveaux algorithmes pour le traitement d'images à N dimensions basés sur des implémentations efficaces pour la détection de structures fines, curvilinéaires et orientées. Nous allons notamment mener notre étude dans le cas 'bruité', c'est à dire dans le cas où les structures sont déconnectées et donc difficiles à détecter (voir Figure A.1 pour des exemples d'applications).
Après un bref rappel de l'état de l'art du filtrage des structures curvilinéaires, fines et orientées en traitement d'images, nous nous focaliserons sur des opérateurs de filtrage non-linéaire issus de la morphologie mathématique appelés ouvertures par chemins. Après avoir conclu sur leurs avantages mais aussi sur leurs limitations, nous allons proposer deux nouveaux opérateurs non-linéaires basés sur les ouvertures par chemins permettant d'accroître leur efficacité notamment dans le cas où les structures sont déconnectées par le bruit et dans le cas de structures dites tortueuses.

A.2 Etat de l'Art du Filtrage des Structures Curvilinéaires

La détection des structures fines et curvilinéaires est encore un challenge en traitement d'images. Depuis 40 ans, un nombre impressionnant de méthodes ont été dédiées à cette tâche dans de nombreux domaines d'applications.
Dans cette étude nous avons décidé de classer ces différentes méthodes en deux groupes : d'une part les filtres cherchant à préserver les structures et d'autre part les filtres cherchant à détecter les structures (voir Figure A.2 pour une illustration).

A.2.1 Filtres Détectant les Structures

Les filtres présentés ici sont fondés sur des modèles de formes et d'intensités faisant l'hypothèse que les structures curvilinéaires présentent un fort contraste dans l'axe orthogonal à leur orientation principale.



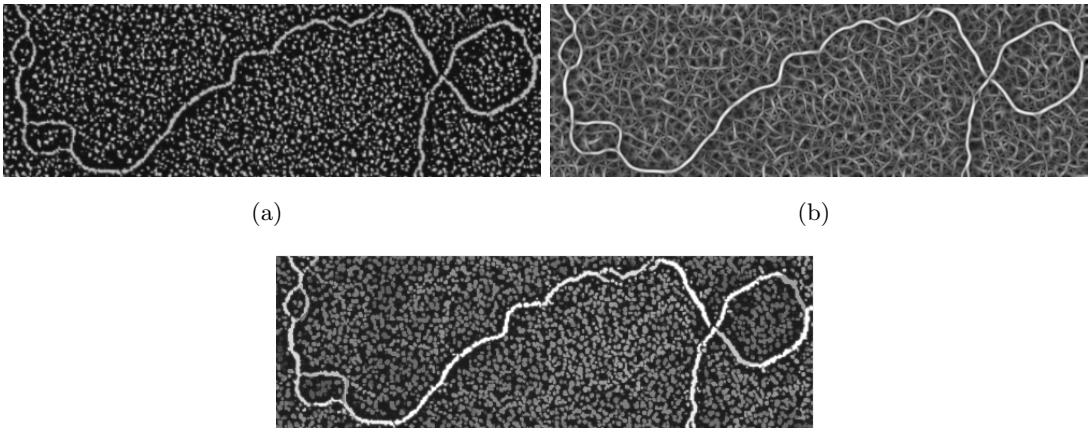
(a) (b)

(c) (d)

Figure A.1: Exemples d'applications: (a).Vaisseaux de la rétine; (b).Route sur image satellitaire; (c).Artères coronaires; (d).Matériau polymère

A.2.1.1 Filtres Orientés

Les filtres orientés peuvent être utilisés dans de nombreuses applications : analyse de texture, détection de bords et de crêtes etc ... L'opération de filtrage orienté consiste prnicipalement à faire varier l'orientation d'un noyau de convolution de base en considèrant une gamme d'orientations possibles et de mesurer par la suite la réponse du



(a) (b)

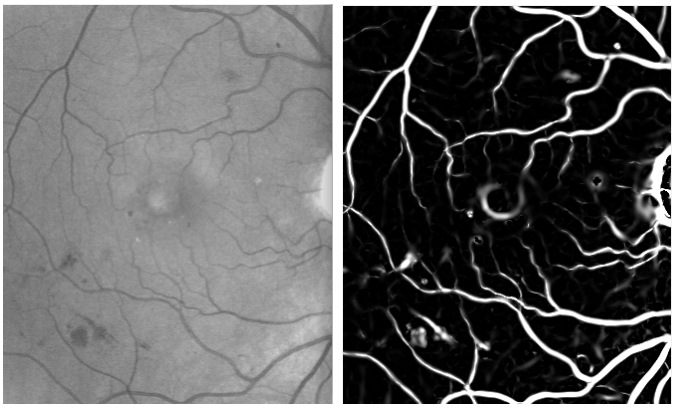
(c)

Figure A.2: Exemples de filtres qui préservent et détectent les structures (a).Image originale (b).Filtre qui détecte; (c).FIltre qui préserve;

contenu de l'image pour chaque orientation. La finalité est donc d'obtenir l'orientation qui correspond à la réponse de magnitude la plus forte.

Dans la famille des filtres orientés, les 'steerable filters' (5), (6) ont des propriétés intéressantes (notamment pour la sélectivité et la complexité algorithmique). En effet la réponse de ces filtres pour un angle donné peut être obtenu par une combinaison linéaire d'un ensemble de filtres de bases.

Un deuxième exemple appartenant à cette famille de filtre est celui des filtres Hessien. En considèrant l'image comme un relief topographique, ces méthodes font l'hypothèse que les structures d'intérêts à détecter sont les lignes de crêtes de ce relief topographique (dans le cas de la détection de structures plus claires que le fond de l'image). Le filtre Hessien (9), (10), (11), (12), (13), (14), est l'un des filtres le plus usités notamment en imagerie médicale. Fondé sur la théorie des espaces d'échelles linéaires (15), les filtres Hessiens sont obtenus en convoluant l'image avec des noyaux de Gaussiennes (dérivées d'ordre un et deux) à différentes échelles en faisant varier l' écart-type du noyau Gaussien suivant une plage de donnée prédéfinie (voir Figure A.3). Même si l'utilisation de ces filtres reste très avantageuse, ils possèdent certaines limitations, notamment une inaptitude à détecter les jonctions.



(a) (b)

Figure A.3: (a).Image originale (b).Filtre Hessien multi-échelle utilisant la mesure de Frangi (14);

A.2.1.2 Chemin de Coût Minimal

L.Vincent (23), (24), a proposé un algorithme de calcul efficace de chemin de coût minimal dans des images 2D en utilisant un connectivité contrainte définissant une orientation donnée. Cet algorithme est particulièrement efficace dans le cas des structures fortement bruitées.
Plus récemment V. Bismuth *et al.* (25) ont généralisé les concepts proposés par Vincent en ajoutant une contrainte de courbure paramètrable sur le chemin propagé.

A.2.2 Filtrage Morphologique

La morphologie mathématique (49), (50), (51), (52), (53), (54) repose sur la théorie des ensembles ce qui fait d'elle un espace de travail 'auto-contenu' et cohérent. Cette théorie est très bien adaptée au traitement d'images car l'information visuelle du monde nous entourant n'est pas translucide mais *a contrario* composée d'objets opaques se chevauchant les uns avec les autres.
Le but de la morphologie mathématique est d'extraire de l'information à partir de transformations non-linéaires appliquées aux objets de l'image.
On pourra notamment chercher des informations à propos de :

- la forme,

- l'orientation,
- la taille,
- la connectivité ...

Dans le cas particulier du filtrage d'éléments fins, curvilinéaires et orientés, ils convient d'adapter la forme de l'élément structurant pour ne pas supprimer toute l'information utile (ce serait le cas par exemple en utilisant un élément structurant en forme de disque).
La puissance des opérateurs de base définis par la morphologie mathématique (l'érosion et la dilatation) est qu'ils peuvent être facilement combinés pour former d'autres opérateurs. On parlera alors d'ouverture pour caractériser la composition d'une érosion suivie d'une dilatation et de fermeture pour caractériser la composition d'une dilatation suivie d'une érosion.
Lorsque l'on applique une ouverture (resp. fermeture) sur une image avec un élément structurant donné, les objets les plus clairs (resp. les plus sombres) ne respectant pas l'arrangement géométrique défini par l'élément structurant seront supprimés.
Dans notre cas d'étude, il vient naturellement l'idée d'utiliser des éléments de type segments orientés. Soille *et al.* (60) proposa un algorithme efficace pour le calcul d'ouvertures et de fermetures sur des lignes de Bresenham. Aprs avoir formulé une version de ces filtres invariante par translation (66), une version robuste au bruit 'rank-based' a été proposée permettant le calcul d'ouvertures et de fermetures sur des segments orientés incomplets (i.e. avec un nombre fixé de pixels manquant r dans un segment de longueur L pixels).
Plus récemment, les ouvertures par chemins furent proposées permettant d'augmenter la flexibilité géométrique des ouvertures par segments. Ces algorithmes sont au coeur de l'étude que nous avons mené au cours de cette thèse.
Le filtrage par attribut constitue une autre branche du filtrage morphologique. Au lieu de comparer les objets de l'image avec des formes prédéinies, les objets sont extraits de l'image et des mesures (appelés attributs) sont calculés en considérant l'object comme une seule composante connexe (75). Ces attributs peuvent par exemple caractériser l'aire, le volume ou encore la forme ou la tortuosité. Nous reviendrons sur le filtrage par attribut dans la partie dédiée aux ouvertures par chemins robustes et sélectives.

A.2.2.1 Discussion

Dans le cas que nous considérons ici, i.e. le cas du filtrage des structures fines, curviliniéaires et orientées dans un contexte bruité, l'étape de prétraitement dont le but est d'augmenter le taux d'information utile par rapport au bruit est critique.
La morphologie mathématique est une théorie qui propose un large de choix de filtres permettant de conserver les propriétés radiométriques et photométriques des structures d'intérêts dans les images tout en réduisant l'impact du bruit.
De plus, un réel effort a été apporté par la communauté scientifique permettant d'obtenir des implémentations efficaces du point de vue des temps de calculs et de l'occupation mémoire. Cela a grandement motivé le travail qui va suivre sur les opérateurs d'ouvertures et de fermetures par chemins.

A.3 Ouvertures par Chemins

A.3.1 Principe

Dans de nombreuses situations, le traitement d'images implique l'utilisation de filtres morphologiques basés sur un élément structurant donné. Ces éléments structurants sont en quelque sorte des sondes que l'on déplace en tout points de l'image pour comparer l'arrangement spatial des objets de l'image avec ceux-ci.
Pour que ces filtres soient efficaces, c'est au praticien en morphologie mathématique de convenir de la forme la plus adaptée à l'application considérée. Dans le cas du filtrage de structures fines, allongées et orientées, il paraît logique de vouloir dans un premier temps utiliser des segments orientés en tant qu'élément structurant.
Malheureusement, les segments orientés ne sont pas suffisant, dans le cas d'applications pratiques, pour s'adapter à la flexibilité locale des structures. De plus, ils peuvent s'avérer être gourmand en terme de temps de calculs en 3D (80).
Pour contrer cette limitation, les ouvertures et fermetures par chemins ont été introduites récemment par Buckley et Talbot dans (82). Peu après, des bases théoriques plus appronfondies furent données avec une implémentation récursive en $O(NL)$ (avec N le nombre de pixels de l'image et L la longueur du chemin en pixels) par les mêmes auteurs. Une implémentation efficace en $O(N \log(L))$ fût introduite par la suite par Talbot et Appleton in (83). Cette implémentation ordonnée est à la base de nos travaux

de recherche.
Plus récemment Cris L. Luengo Hendriks (81) a proposé une modification de l'implémentation des ouvertures par chemins permettant sa généralisation au cas N dimensionnel. C'est cette implémentation que nous allons étudier ici.

A.3.2 Aspects Théoriques

Les détails des aspects théoriques peuvent être trouvés dans (1). Ici, nous allons juste rappeler les grands principes qui se trouvent derrière ces opérateurs.

A.3.2.1 Graphe d'Adjacence, Dilatation et Ouvertures par Chemins Binaires

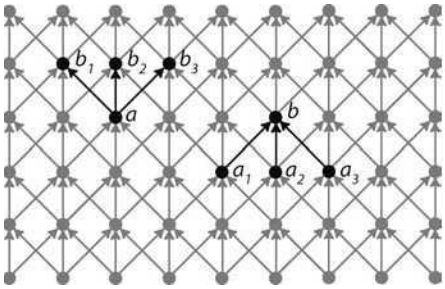


Figure A.4: Illustration d'un graphe d'adjacence défini par une relation binaire;

Partant d'un ensemble de points de E (E correspondant au support spatial de l'image), on définit un graphe d'adjacence (e.g. Figure A.4) entre ces points à partir d'une relation binaire $x \to y$. Grâce au graphe d'adjacence défini ci-dessus, on peut définir un δ-chemin de longueur L qui est un L-tuple:

$$\boldsymbol{a} = (a_1, a_2, ..., a_L) \text{ si } a_k \to a_{k+1} \tag{A.1}$$

On note $\Pi_L(X)$ l'ensemble des δ-chemin de longueur L dans un sous-ensemble X de E par:

$$\Pi_L(X) = \{\boldsymbol{a} \in \Pi_L : \sigma(\boldsymbol{a}) \subseteq X\} \tag{A.2}$$

avec $\sigma(\boldsymbol{a}) = \sigma(a_1, a_2, ..., a_L)$ l'ensemble des éléments du chemin. Les ouvertures par chemins sont donc définies par:

$$\alpha_L(X) = \bigvee\{\sigma(\boldsymbol{a}) : \boldsymbol{a} \in \Pi_L(X)\} \tag{A.3}$$

$\alpha_L(X)$ respecte les propriétés canoniques algébriques d'une ouverture, i.e : croissance, anti-extensivité et idempotence.
Ici nous venons de définir les ouvertures par chemins, mais bien entendu cette définition s'applique aussi aux fermetures par chemins définies par complémentation (par dualité des opérateurs d'ouvertures et de fermetures).

A.3.2.2 Extension aux Images à Niveaux de Gris

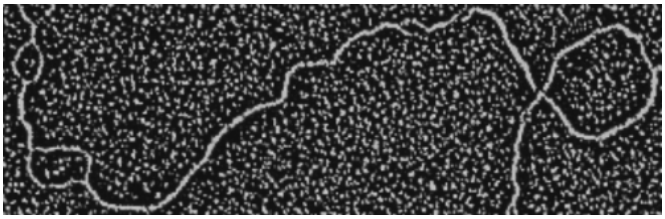
L'extension de l'opérateur binaire des ouvertures par chemins a été proposée en se basant sur le principe de superposition de seuils (52), (84).
L'ouverture par chemins en niveaux de gris d'une image I est alors définie par :

$$\Pi_L^t(I) = \Pi_L(X_t(I)) \tag{A.4}$$

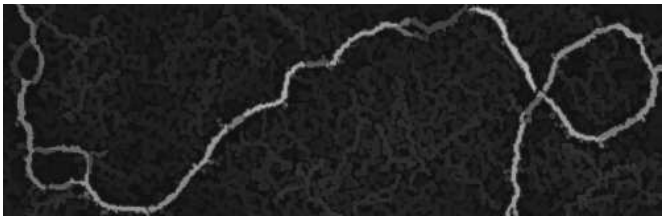
où $X_t(I) = \{x \in E : I(X) \geq t\}$ est le '*level set*' de I à la valeur t.
Cela revient à dire qu'une ouverture (resp. fermeture) par chemins en niveaux de gris donne pour chaque pixel de l'image la plus haute (resp. la plus basse) valeur pour laquelle l'ouverture (resp. la fermeture) par chemins binaire est *vrai*. Si on représente une image à niveaux de gris comme un empilement d'ensembles binaires d'éléments, appliquer une ouverture par chemins en niveaux de gris revient à l'appliquer sur chacun des ensembles binaires et à les empiler á nouveau (voir illustration Figure A.5).

A.3.3 Algorithme

Nous allons décrire le principe de l'algorithme d'ouvertures par chemins proposé par Cris Luengo Hendriks (81) applicable sur des images à N dimensions.
Cet algorithme est fondé sur l'implémentation ordonnée proposée précédemment par Talbot et Appleton (83).
Pour appliquer une ouverture (resp. fermeture) par chemins, l'algorithme commence par créer un tableau linéaire d'adresses en mémoire, ordonné de manière croissante (resp. décroissante) par rapport aux valeurs des pixels à qui elles se réfèrent. En commençant du niveau le plus bas au niveau le plus haut, chaque pixel qui est encore actif sera traité de manière indépendante et servira de point de départ à un algorithme de propagation de longueurs (en amont et en aval de ce pixel) afin de mettre à jour les longueurs des pixels actifs de l'image.
Deux images temporaires sont créées (image longueur amont et aval : λ^+ et λ^-) et



(a)



(b)

Figure A.5: (a).Image originale; (b).Ouverture par chemins $L = 60$;

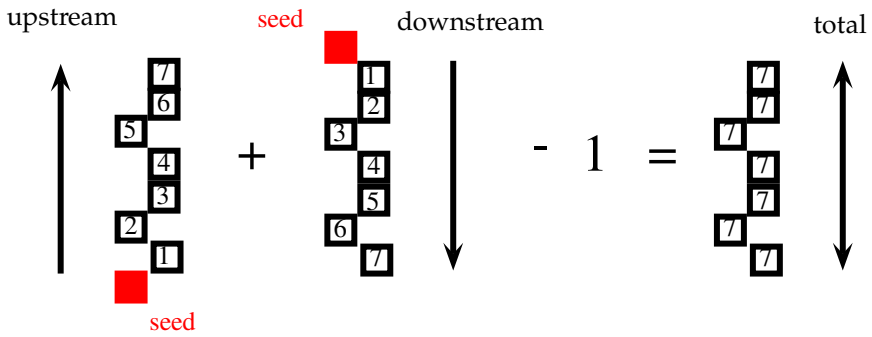


Figure A.6: Illustration du principe de propagation au coeur de l'algorithme d'ouvertures par chemins.

permettent d'accumuler les longueurs obtenues pour chaque pixel (voir A.6 pour une illustration du processus de propagation). La propagation du chemin est gérée par une première pile FIFO (*First In First Out*) de manière récursive jusqu'à ce que l'on atteigne un maximum de L pas de propagations élémentaires ou lorque plus aucun pixels actifs voisins du pixel courant de la pile de propagation n'est trouvé.
Lorsque la valeur de la longueur de chemin 'amont' ou 'aval' d'un pixel est modifiée, ce pixel est chargé dans une deuxième pile FIFO, puis la valeur totale du chemin

parcourant ce pixel est calculée de la manière suivante :

$$\lambda = \lambda^+ + \lambda^- - 1 \tag{A.5}$$

Si λ passe en dessous de la valeur L, alors ce pixel est désactivé, les valeurs des longueurs 'amont' et 'aval' sont remises à 0 et sa valeur en sortie sera égale à la valeur courante du seuil (c'est à dire, à la valeur du pixel pris comme point de départ).
Notons que l'indépendance de cet algorithme à la dimension de l'image est gérée de manière simple en créant un tableau linéaire correspondant aux adresses des pixels. L'accès aux voisins d'un pixel se fait en ajoutant un décalage linéaire en mémoire.
D'un point de vue utilisateur, il est usuel d'exécuter les ouvertures par chemins en considérant plusieurs orientations principales et de faire l'union des réponses sur chaque orientation. En 3D par exemple, nous avons choisi d'utiliser des sous-ensembles de la 26 connexité pour générer les résutats. En pratique nous avons considéré 7 orientations principales donnant un bon compromis entre isotropie du résultat final et temps de calculs (voir Figure A.7).

A.3.4 Limitations

Les ouvertures par chemins sont très efficaces pour détecter des structures curvilignes orientées et qui peuvent aussi être flexibles localement. Malgré cela, les ouvertures par chemins présentent une grande sensibilité au bruit générant des déconnexions dans les structures d'intérêts.
Dans (83), les ouvertures par chemins incomplets ont été introduites pour résoudre cette limitation. En autorisant un nombre limité de K pixels n'appartenant pas à l'ensemble de propagation dans un chemin de longueur L, la robustesse, autement dit, la flexibilité radiométrique est largement améliorée (voir Figure A.8).
Cependant, certains problèmes se posent en ce qui concerne l'implémentation des ouvertures par chemins incomplets. En effet, l'occupation mémoire et la complexité algorithmique est accrue de manière significative. Cela représente un challenge pour le développement d'une version 3D.
Un deuxième point important est que le paramètre de robustesse K est typiquement un paramètre croissant avec L. En effet il est logique que plus on cherche des chemins de longueurs importantes et plus la probabilité augmente de rencontrer certaines déconnexions dans les chemins d'où une augmentation mécanique de K.

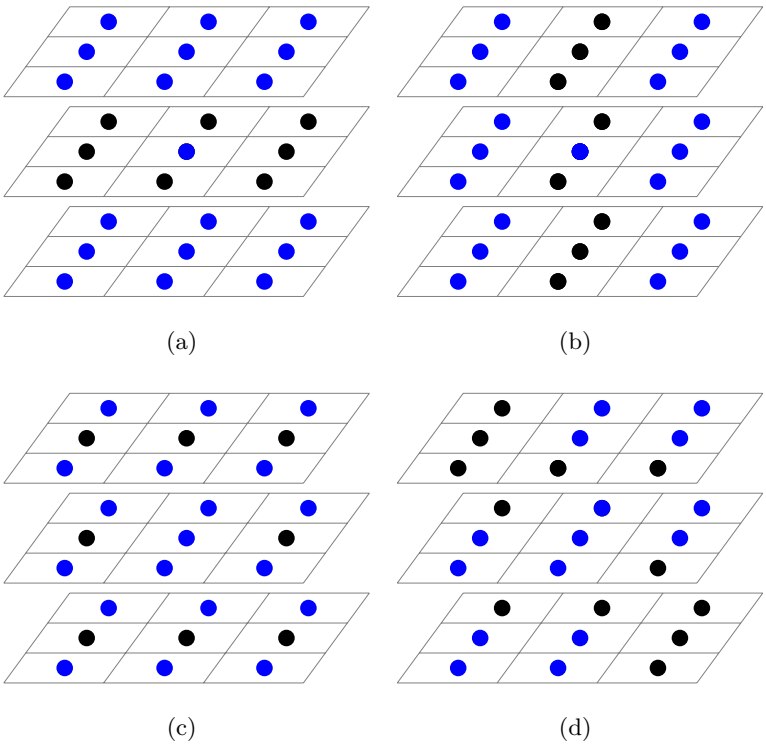


Figure A.7: Exemples de graphes d'adjacences en 3D; (a), (b) et (c) sont orientés selon les axes principaux, (d) selon une des diagonales de l'image

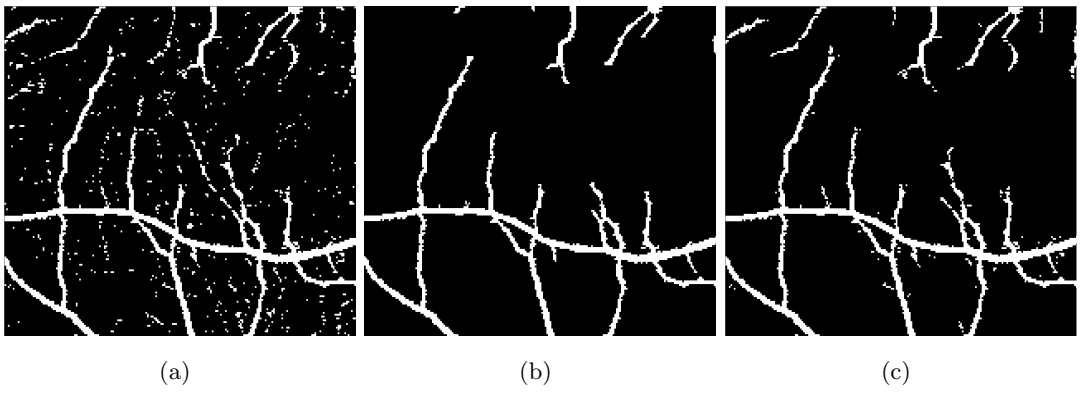


Figure A.8: Détection de vaisseaux en 2D (a).Transformation 'TOP HAT' anisotropique (image d'entrée); (b).Résultat de l'ouverture par chemins ($L = 45$); (c).Résultat de l'ouverture par chemins incomplets($L = 45$,$K = 2$);

A.4 Ouvertures par Chemins Robustes

A.4.1 Motivations et Spécifications

Nous venons de voir que les ouvertures et fermetures par chemins étaient très efficaces pour discriminer les structures curvilinéaires, fines, localement flexible des autres structures (e.g. bruit compact). Cependant, à la lumière des limitations évoquées précédemment, notamment par rapport à la difficulté de prendre en compte les déconnexions au sein de la propagation d'un chemin dans une implémentation efficace, nous proposons de définir un nouveau cadre de travail.
Malgré tout, nous devons respecter certaines contraintes :

- l'algorithme devra être indépendant de la dimension
- la complexité algorithmique devra être réduite
- l'occupation mémoire devra être constante
- le paramètre permettant de gérer la robustesse ne devra pas être dépendant de la longueur du chemin.

A.4.2 Proposition

Considérant les spécifications énoncées ci-dessus, nous proposons de construire un algorithme indépendant de la dimension de l'image : les ouvertures par chemins robustes (85). Cet algorithme sera basé sur celui des ouvertures par chemins complets indépendants de la dimension de l'image proposé par Cris Luengo Hendriks(81).
Pour chaque couple de paramètres (L, G), une famille plus grande de chemins incomplets et flexibles est générée.
Le principe est le suivant : à partir d'une longueur maximale de chemin G en pixels, la propagation du chemin sera autorisée entre deux éléments d'un chemin incomplet si la longueur de déconnexion entre ces deux éléments est inférieure ou égale à G. Cette simple proposition pourra générer une implémentation dont l'occupation mémoire sera constante et dont la complexité sera réduite.

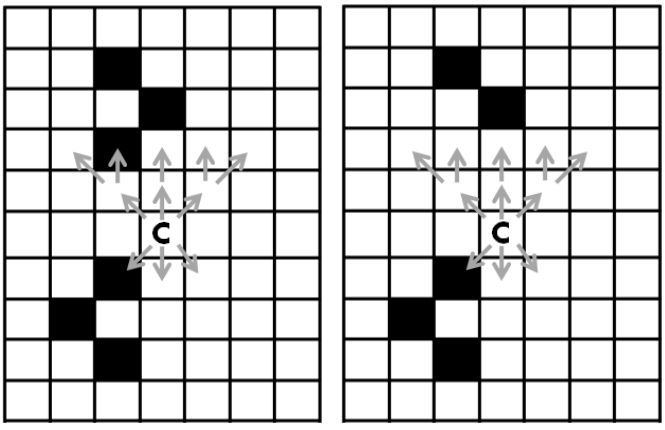
A.4.3 Aspects Théoriques

Nous allons donner les aspects théoriques relatifs aux ouvertures par chemins robustes. A partir d'une longueur donnée G, en pixels, la propagation du chemin sera autorisée si la longueur entre deux éléments d'un chemin ne dépasse pas G ($1 \leq G < L$). Les pixels appartenant à ces déconnexions seront appelés 'noise' pixels.
Dès lors, nous pouvons définir la condition pour un pixel dans E d'être un 'noise' pixel : NPC_k. For $k = 1...G$ nous avons, pour un chemin $\boldsymbol{a}$ de longueur k dans X^c (Figure A.9 donne un exemple pour $G = 2$):

$$\mathrm{NPC}_k(\boldsymbol{a}) = \begin{cases} \check{\delta}(\{a_1\}) \cap X \neq \emptyset \\ \delta(\{a_k\}) \cap X \neq \emptyset \end{cases} \tag{A.6}$$

a_1 est le premier élement du chemin et a_k le dernier dans X^c.
D'où la définition de $\Pi^G_{\mathrm{noise}}(X)$ comme étant :

$$\Pi^G_{\mathrm{noise}}(X) = \{\boldsymbol{a} \in \bigvee_{k=1..G} \Pi_k(X^c) : \boldsymbol{a} \;\; satisfy \;\; \mathrm{NPC}_k\} \tag{A.7}$$



(a) (b)

Figure A.9: Recherche des 'noise' pixels pour $G = 2$ (a).Le candidat **C** est marqué; (b).Le candidat **C** n'est pas marqué.

Maintenant nous pouvons définir les G-robust δ-chemin de longueur L For $1 \leq G < L$

$$\Pi^G_L(X)_{\mathrm{Robust}} = \Pi_L(X \cup X^G_{\mathrm{noise}}) \tag{A.8}$$

avec

$$X^G_{\text{noise}} = \{x \in \sigma(\boldsymbol{a}_{\text{noise}}) : \boldsymbol{a}_{\text{noise}} \in \Pi^G_{\text{noise}}(X)\} \tag{A.9}$$

$\sigma(\boldsymbol{a}_{\text{noise}})$ est défini de manière analogue comme étant l'ensemble des points appartenant à un chemin de $\Pi^G_{\text{noise}}(X)$. Ainsi on peut définir les ouvertures par chemins robustes binaires comme étant:

$$\alpha^G_L(X)_{\text{Robust}} = \bigvee\{\sigma(\boldsymbol{a}) \cap X : \boldsymbol{a} \in \Pi^G_L(X)_{\text{Robust}}\} \tag{A.10}$$

Cela revient à dire que les ouvertures par chemins robustes sont équivalentes aux ouvertures par chemins complets en considèrant une extension de l'ensemble de propagation (1). Cet opérateur respecte les propriétés canoniques d'une ouverture, c'est-à-dire, la croissance, l'anti-extensivité et l'idempotence (la preuve est directe car l'opérateur associant à un ensemble donné son ensemble étendu est une dilatation).

A.4.4 Algorithme

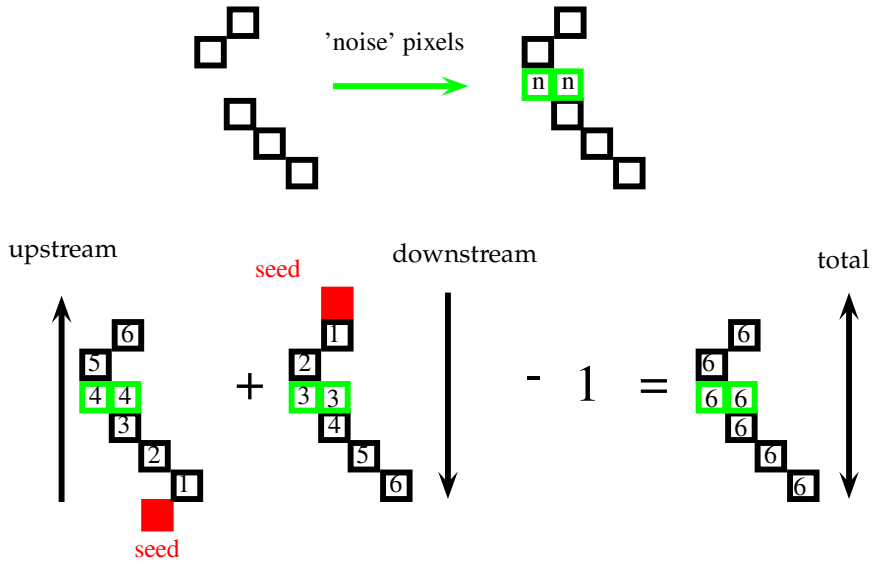


Figure A.10: Principe de l'algorithme des ouvertures par chemins robustes

L'algorithme des ouvertures par chemins robustes est basé sur celui proposé par Cris L. Luengo Hendriks. Nous décrirons donc uniquement les modifications apportées

pour robustifier cet algorithme.
La première étape supplémentaire est l'algorithme de recherche de déconnexions dans la limite de G pixels entre deux parties d'un chemin. Celui-ci est réalisé en cherchant de manière récursive des pixels 'actifs' en considèrant les pixels du fond (ou encore pixels venant d'être désactivés après un changement de seuil) de l'image comme point de départ (voir Figure A.10.) Si la longueur du chemin reliant les deux extrémités est inférieure à G, alors, le pixel point de départ de la recherche est marqué comme étant 'noise' pixel, c'est-à-dire, comme appartenant à une déconnexion. Notons que les longueurs de chemins propagées, amont et aval ne sont pas remises à 0 car ce pixel va servir pour la suite lors de la propagation des chemins.
La deuxième modification majeure intervient dans la fonction de propagation qui doit être redéfinie pour prendre en compte maintenant les pixels appartenant aux déconnexions du chemin.
Pour respecter le principe de superposition de seuils, il convient aussi de mettre en place un processus de mise à jour des 'noise' pixels lorsque l'on passe d'un seuil à l'autre. En effet, il faut marquer les nouveaux 'noise' pixels mais aussi re-vérifier le status des 'noise' pixels des seuils précédents. Si il arrivait qu'un pixel perdait son status de 'noise' pixel, il faudrait relancer une procédure de propagation à partir de celui-ci.

A.4.5 Temps de Calculs et Résultats Expérimentaux

Nous allons présenter les principaux résultats des ouvertures par chemins robustes en les comparant avec ceux des ouvertures par chemins complets et incomplets.
Améliorer la détection des structures fines et peu contrastées tout en réduisant le taux de fausses détections est le principal but de l'extraction des vaisseaux sanguin de la rétine. Ce problème a été largement abordé depuis plusieurs années (89), (90). Nous allons donc comparer les résultats de la détection des vaisseaux de manière qualititative ainsi que les temps de calculs.
Après avoir extrait le canal 'vert' de l'image RGB de la rétine (donnant le meilleur contraste), une transformation en chapeau haut de forme est appliquée dans le but d'extraire les minima locaux de l'image. L'image résultante contient beaucoup de bruit, spots et déconnexions des vaisseaux.
La figure A.11 montre le résultat du seuillage des différentes images filtrées ainsi que

de l'image originale. En utilisant les ouvertures par chemins complets, on peut facilement remarquer que la totalité du bruit est supprimé, cependant, on notera que cet opérateur n'est pas capable de détecter des vaisseaux présentant des déconnexions. Les ouvertures par chemins incomplets donnent des résultats meilleurs, sous réserve que les déconnexions ne dépassent pas un certain seuil (en effet si le nombre de déconnexions est supérieur à K, la propagation du chemin est stoppée). L'utilisation des ouvertures par chemins robustes donne le meilleur résultat en terme de détection. Notons aussi la présence de certains artefacts (absents dans le cas des ouvertures par chemins complets et moins présents dans le cas incomplets) résultant de la corrélation de pixels appartenant au bruit disposés de manière aléatoire et formant un arrangement valide.
La Figure A.12 compare les temps de calculs des différents algorithmes d'ouvertures par chemins en utilisant la même image d'entrée. Comme attendu, la complexité algorithmique des ouvertures par chemins robustes est largement réduite par rapport à celle des ouvertures par chemins incomplets dans tout les cas. Par exemple, pour $G = 2$ et $K = 2$, les ouvertures par chemins robustes sont 7 fois plus rapides que les ouvertures par chemins incomplets. De plus, l'occupation mémoire des ouvertures par chemins robustes est constante par rapport à G. Nous observons que les temps de calculs des ouvertures par chemins robustes sont linéaires par rapport à G (environ 0.4×G).

A.5 Ouvertures par Chemins Robustes Sélectives

A.5.1 Tortuosité et Opérateurs par Chemins

Dans les sections précédentes, nous avons introduit les ouvertures par chemins ainsi que leurs implémentations efficaces.
Dans cette section nous voulons mettre en évidence une limitation des ouvertures par chemins (complets ou robustes) qui a été récemment soulevée par V. Morard *et al.* dans (76) and (77). En effet, du fait de l'utilisation d'un graphe de connexité contraint par une orientation donnée, les ouvertures par chemins sous-estiment clairement la longueur des structures dites tortueuses. La Figure A.13 en est un bon exemple. Pour répondre à cette problématique nous allons proposer un algorithme qui permet de combiner les ouvertures par chemins robustes et la reconstruction morphologique des parties tortueuses des structures. Pour se faire nous allons tout d'abord rappeler

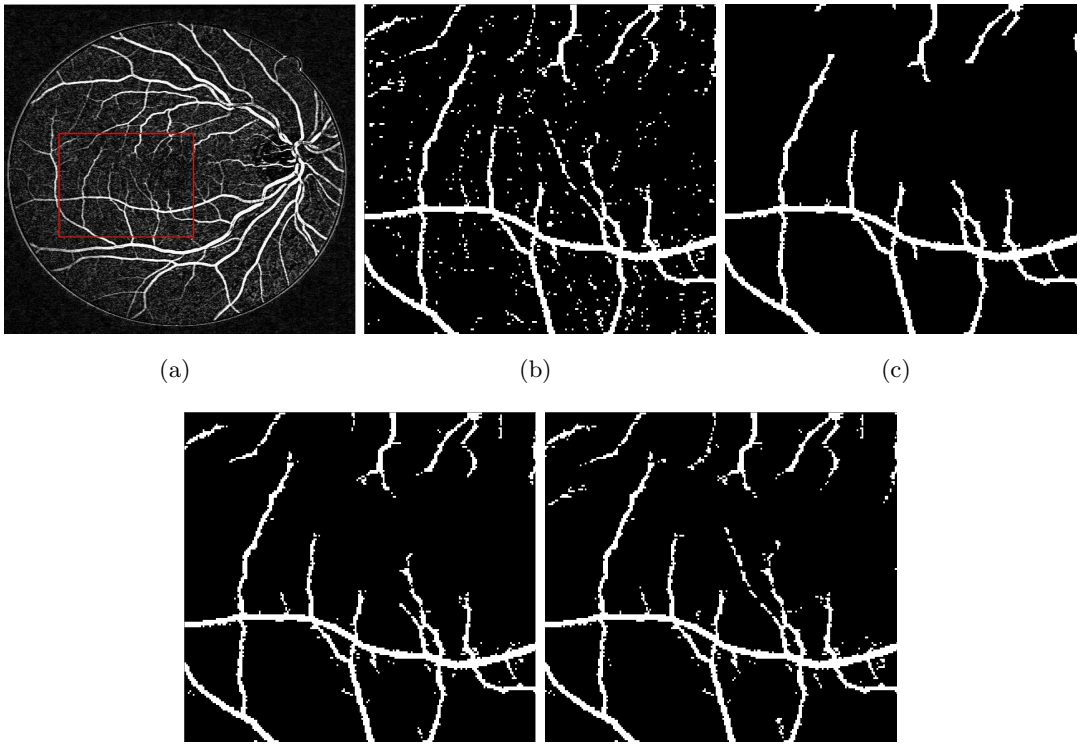
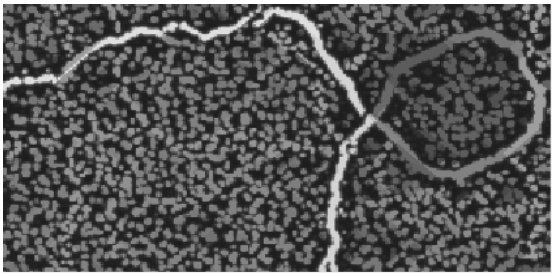


Figure A.11: Détection des vaisseaux de la rétine (2D) (a). Image d'entrée; (b). Image d'entrée seuillée; (b). Résultat des ouvertures par chemins complets ($L = 45$); (c). Résultat des ouvertures par chemins incomplets ($L = 45$, $K = 2$); (d). Résultat des ouvertures par chemins robustes ($L = 45$, $G = 2$); le même seuil a étéappliqué sur (b),(c),(d) et (e).

	COPO	RPO	IOPO	RPO	IOPO	RPO	IOPO
Tolerance	$K = 0$	$G = 1$	$K = 1$	$G = 2$	$K = 2$	$G = 3$	$K = 3$
L=10	0.71	1.03	5.61	1.34	8.78	1.81	12
L=50	0.79	1.03	5.93	1.33	9.31	1.77	12.93
L=100	0.82	1.05	6.03	1.35	9.47	1.78	13.2

Figure A.12: Temps de calculs des différentes impléentations des ouvertures par chemins. La taille de l'image est 565x585x8-bit.

comment caractériser la tortuosité des structures par l'usage de filtres s'appliquant sur les composantes connexes.



(a)

Figure A.13: Exemple de l'incapacité des ouvertures par chemins à détecter des structures dites tortueuses (RPO ($L = 150, G = 2$)).

A.5.2 Filtrage par Attributs Géodésiques

A.5.2.1 Filtrage par Attributs

Soit $X_i \in E$ une compososante connexe de l'image, un filtrage par attribut opère une transformation non-linéaire de l'image conduite par le calcul d'un prédicat binaire C_λ. Ce critère vérifie si un attribut calculé sur la composante connexe X_i est supérieur ou non à un seuil donné.

Un opérateur par attribut binaire Att_λ est défini comme suit:

$$Att_\lambda(X_i) = \begin{cases} X_i & \text{if } X_i \text{ satifies the criterion } C_\lambda \\ \emptyset & \text{otherwise} \end{cases} \tag{A.11}$$

Basé sur cette définition, (96) proposa l'amincissement par attribut ρ^{Att_λ}:

$$\rho^{Att_\lambda} = \bigvee_X Att_\lambda(X_i) \tag{A.12}$$

Les amincissements par attributs sont idempotents et anti-extensifs. Cependant la croissance de l'opérateur n'est pas assurée. Si oui, on parlera alors d'ouverture par attribut.

A.5.2.2 Filtrage par Attributs

Définie par Lantuejoul et Maisonneuve (95), la distance géodésique d'une composante connexe (ou objet) fournit une caractérisation de sa longueur.
Considèrant deux points x et y appartenant à un objet X, la distance géodésique est

la longueur de l'arc le plus court.
Sur X le diamètre géodésique est égal à:

$$L(X) = \sup_{(x,y)\in X} dX(x,y) \tag{A.13}$$

avec $dX(x,y)$ correspondant à la distance géodésique entre x et y.
Les principaux atouts de l'utilisation du diamètre géodésique sont les suivants:

- sa définition est générale et peut s'appliquer aux objets de formes quelconques,
- sa définition est robuste aux petits changements dans la forme de l'objet,
- son calcul peut servir à dériver d'autres attributs.

A partir du diamètre géodésique, (77) et (76) proposa de dériver un autre attribut géodésique, la tortuosité géodésique τ.
τ est défini comme étant le rapport entre le diamètre géodésique et la distance Euclidienne entre les deux extrémités géodésiques qui la minimise.
Soit $PE(X)$, l'ensemble des paires extrémités géodésiques, nous avons :

$$L_{Eucl}(X) = \min_{\{x,y\}\in PE(X)} ||x,y|| \tag{A.14}$$

avec $||.||$ la distance Euclidienne dans $\mathbb{Z}^2$.
La tortuosité est définie par:

$$\tau(X) = \frac{L(X)}{L_{Eucl}(X)} \tag{A.15}$$

A.5.3 Ouvertures par Chemins Robustes Sélectives

A.5.3.1 Principe

Les ouvertures par chemins robustes sélectives sont basées sur le principe de la reconstruction morphologique à niveaux de gris (99).
Le principe est d'utiliser une carte de tortuosité comme image masque pour reconstruire les parties des structures filtrées par les ouvertures par chemins robustes A.14.

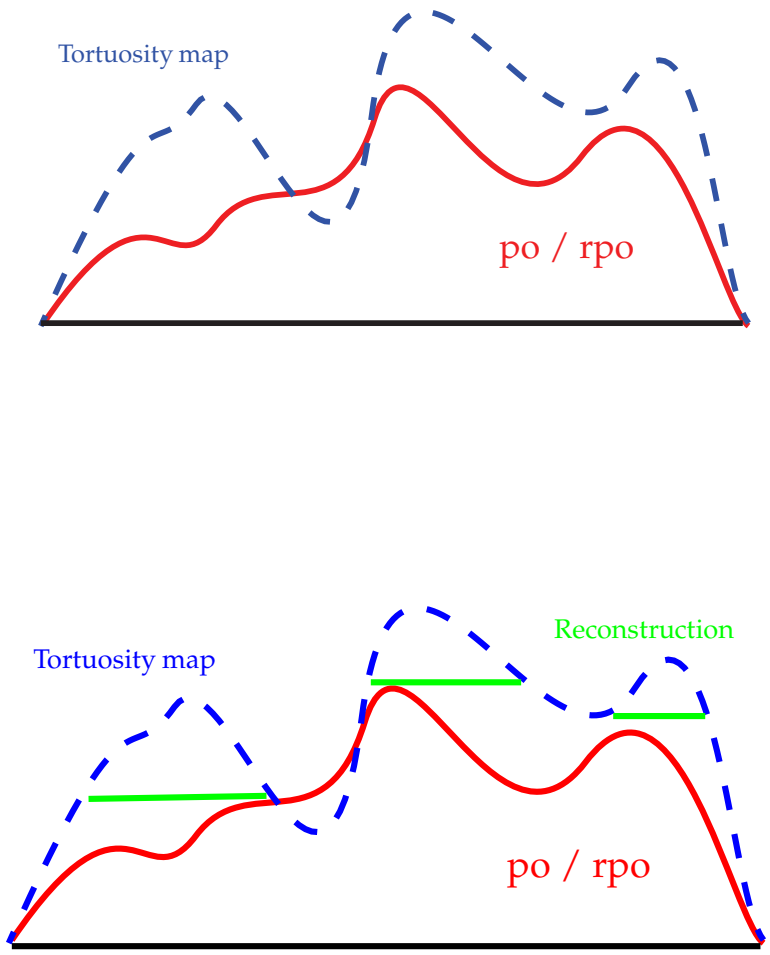


Figure A.14: Principe des ouvertures par chemins robustes sélectives

A.5.3.2 Aspects Théoriques

Rappelons que nous pouvons définir l'ensemble des G-robust chemins de longueur L dans $X \subseteq E$ comme:

$$\alpha_L^G(X)_{\text{Robust}} = \bigvee \{\sigma(\boldsymbol{a}) \cap X : \boldsymbol{a} \in \Pi_L^G(X)_{\text{Robust}}\} \tag{A.16}$$

Avec une légère modification nous pouvons prendre en compte la connectivité à travers les pixels 'noise' dans E:

$$\alpha_L^G(X)_{\text{Robust}\,E} = \bigvee\{\sigma(\boldsymbol{a}) : \boldsymbol{a} \in \Pi_L^G(X)_{\text{Robust}}\} \tag{A.17}$$

Ouvertures par chemins robustes sélectives Les ouvertures par chemins robustes et sélectives sont définies comme étant le résultat de la reconstruction morphologique du résultat des ouvertures par chemins dans l'union du résultat des ouvertures par chemins et de la carte de tortuosité.

$$\boldsymbol{\alpha}_L^T(X) = \rho_{(\alpha_L^G(X)_{\text{Robust}\,E} \vee T)} \alpha_L^G(X)_{\text{Robust}\,E} \tag{A.18}$$

A.5.3.3 Algorithme

L'algorithme des ouvertures par chemins robustes et sélectives utilise le principe de superposition de seuils pour combiner en une implémentation les ouvertures par chemins robustes et la reconstruction morphologique.
A un seuil donné T, à la fin de la procédure de mise à jour, une fois que la proc

A.5.3.4 Résultats

Dans cette section nous allons présenter des exemples d'applications des ouvertures par chemins robustes et slectives pour l'imagerie biomdicale 2D et 3D.
Le premier exemple (voir Figure A.15) illustre leurs usages pour la dtection d'un brin d'ADN en imagerie microscopique biomdicale. Il est clair que les ouvertures par chemins (complets et incomplets) sous-estiment le longueur des parties tortueuses de l'image (cela est du à au fait qu'ils sont construits sur un graphe constraint et orienté) alors que la tortuosité géodésique détecte très bien ces parties.
la combinaison de ces deux oprateurs, les ouvertures par chemins robustes slectives donne les meilleurs résultats pour la détection des structures allongées, bruitées mais aussi tortueuses.
Notons que sur cette image (Figure A.15), de taille 500×160×8-bit, l'union sur quatre orientations est exécutée en moins de une seconde. Un autre exemple d'applications est donné en 3D pour l'extrations des vaisseaux dans une image d'angiographie (voir Figure A.16). Les temps de calculs pour cette image de taille 256×256×256×8-bit est de 9.5s pour une orientation.

A.6 Conclusions et Perspectives

Nos travaux de recherche se sont portés principalement sur l'étude des ouvertures et fermetures morphologiques par chemins pour des applications de traitement d'images pour l'imagerie 3D.

Nous avons proposé deux contributions théoriques originales ainsi que des algorithmes efficaces pour permettre leurs implémentations dans le but de répondre aux principales limitations de ces opérateurs, notamment le manque de robustesse aux déconnexions des structures fines générées par le bruit présent dans l'image et aux structures dites tortueuses.

Nous avons proposé les ouvertures par chemins robustes, qui, grâce à la donnée d'une longeur de déconnexion G, permettent d'élargir l'ensemble de propagation des chemins et de robustifier l'algorithme. Les temps de calculs en 2D et 3D et le fait d'avoir une implémentation à mémoire constante permettent sans problème une utilisation pratique sur des grands volumes de données (on pourrait notamment paralléliser avec openMP de manière très simple l'algorithme car le calcul sur chaque orientation est indépendant).

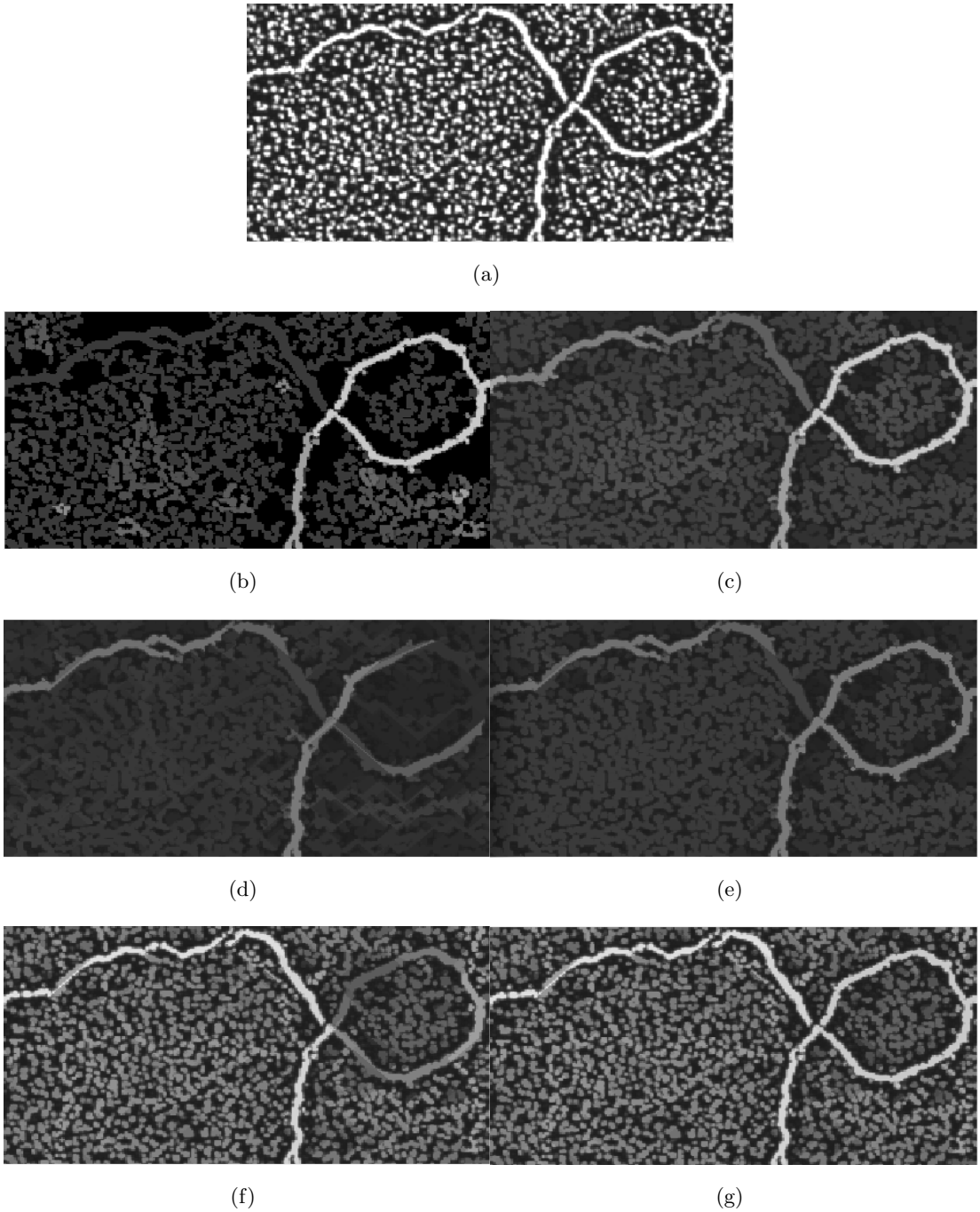
Pour répondre à la deuxième limitation principale des ouvertures par chemins (complets et robustes), la difficulté de détecter les structures dites tortueuses, nous avons proposé un algorithme basé sur la combination de l'algorithme des ouvertures par chemins robustes et de reconstruction géodésique : les ouvertures par chemins robustes et sélectives. L'intérêt est alors d'utiliser la réponse d'un filtre par attribut géodésique, la tortuosité géodésique, en tant que masque pour reconstruire les structure dites tortueuses.

En terme d'applications nous avons proposé une méthode complète de prétraitements pour le rehaussement des vaisseaux sanguins de la rétine en 2D et nous avons prouvé de manière qualitative (courbes ROC) la supériorité des opérateurs morphologiques par chemins par rapport aux filtres communément utilisés.

Même si nous avons fourni des implémentations efficaces de nouveaux filtres morphologiques, il y a un besoin d'accélérer constamment les temps de calculs de méthodes qui deviennent de plus en plus complexes. Un bon moyen d'accélérer les algorithmes peut se trouver dans des stratégies de parallelisation massive.

Dans ces travaux nous avons choisi de faire une séparation entre les méthodes servant à détecter et à préserver les structures dans le but de les classer. Cependant des futurs

travaux pourraient être consacrés à essayer de combiner les approches multi-échelles (filtre de type Hessien basé sur les dérivées secondes de noyaux Gaussiens) avec les ouvertures par chemins robustes. On peut noter dans la litérature une combinaison de ce type avec des ouvertures par segments orientés dans (71).



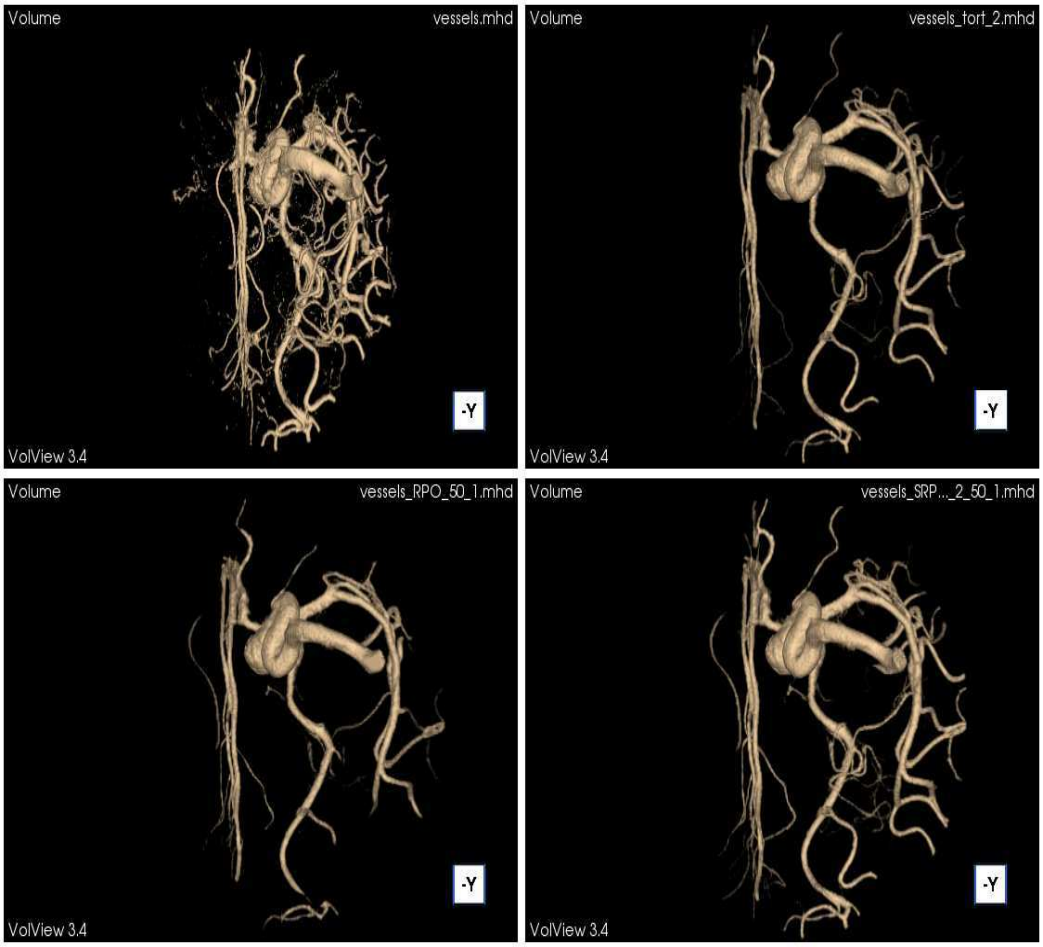
(a)

(b) (c)

(d) (e)

(f) (g)

Figure A.15: Résultats et comparaisons en 2D; (a).Image d'entrée; (b).Tortuosité géodésique ($\tau = 3$); (c).Diamètre barycentrique ($L = 150$); (d).Ouvertures par chemins complets ($L = 100$); (e).Ouvertures par chemins robustes($L = 150$, $G = 2$); (f).Ouvertures par chemins robustes sélectives($L = 150$, $G = 2$ et $\tau = 3$).



(a)

Figure A.16: Résultats et comparaisons en 3D; De gauche à droite et de haut en bas: Rendu surfacique de : l'image originale, tortuosité géodésique, RPO ($(L, G) = (50, 1)$) et SRPO.

References

[1] H. Heijmans, M. Buckley, and H. Talbot. **Path openings and closings**. *Journal of Mathematical Imaging and Vision*, **22**(2):107–119, 2005. 2, 24, 28, 40, 112, 119

[2] R Poli and G Valli. **An algorithm for real-time vessel enhancement and detection**. *Comput Methods Programs Biomed*, **52**(1):1–22, 1997. 6

[3] Dietmar Kunz and Bernhard Schweiger. **Line Detection in Strongly Noise-Corrupted Images.** In Hans-Peter Meinzer, Heinz Handels, Alexander Horsch, and Thomas Tolxdorff, editors, *Bildverarbeitung fr die Medizin*, Informatik Aktuell, pages 50–54. Springer, 2005. 7

[4] Vincent Bismuth, Laurence Vancamberg, and Sebastien Gorges. **A comparison of line enhancement techniques: applications to guide-wire detection and respiratory motion tracking**. pages 72591M–72591M–9, 2009. 7

[5] William T. Freeman and Edward H. Adelson. **The Design and Use of Steerable Filters**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**:891–906, 1991. 7, 8, 56, 108

[6] M. Jacob and M. Unser. **Design of Steerable Filters for Feature Detection Using Canny-Like Criteria**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(8):1007–1019, August 2004. 7, 8, 56, 108

[7] F. John Canny. **A Computational Approach to Edge Detection**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **8**(6):679–698, 1986. 8

[8] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. **Multiscale vessel enhancement filtering**. pages 130–137. Springer-Verlag, 1998. 9, 55

[9] A. F. Frangi, W. J. Niessen, R. M. Hoogeveen, T. V. Walsum, and M. A. Viergever. **Model-Based Quantitation of 3D Magnetic Resonance Angiographic Images**. *IEEE Trans. Med. Imaging*, **18**(10):946–956, 1999. 9, 108

[10] A. F. Frangi, W. J. Niessen, P. J. Nederkoorn, J. Bakker, W. P. Mali, and M. A. Viergever. **Quantitative analysis of vascular morphology from 3D**

MR angiograms: In vitro and in vivo results. *Magnetic Resonance in Medecine*, **45**(2):311–22, 2001. 9, 108

[11] Yoshinobu Sato, Shin Nakajima, Hideki Atsumi, Thomas Koller, Guido Gerig, Shigeyuki Yoshida, and Ron Kikinis. **3D Multi-scale line filter for segmentation and visualization of curvilinear structures in medical images**. In *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery*, CVRMed-MRCAS '97, pages 213–222, 1997. 9, 108

[12] C. Lorenz, I.-C. Carlsen, T.M. Buzug, C. Fassnacht, and J. Weese. **A multi-scale line filter with automatic scale selection based on the Hessian matrix for medical image segmentation**. In Bart Haar Romeny, Luc Florack, Jan Koenderink, and Max Viergever, editors, *Scale-Space Theory in Computer Vision*, **1252** of *Lecture Notes in Computer Science*, pages 152–163. Springer Berlin Heidelberg, 1997. 9, 108

[13] Hidenori Shikata, Eric A. Hoffman, and Milan Sonka. **Automated segmentation of pulmonary vascular tree from 3D CT images**. pages 107–116, 2004. 9, 108

[14] L. Antiga. **Generalizing vesselness with respect to dimensionality and shape**. 08 2007. 9, 10, 55, 108, 109

[15] Jan J. Koenderink. **The structure of images**. *Biological Cybernetics*, **50**(5):363–370–370, August 1984. 9, 108

[16] S. D. Olabarriaga, M. Breeuwer, and W. J. Niessen. **Minimum Cost Path Algorithm for Coronary Artery Central Axis Tracking**. In *in CT Images, MICCAI 2003, LNCS 2879*, pages 687–694, 2003. 9, 12

[17] E. Candes and D. Donoho. **Curvelets: A Surprisingly Effective Nonadaptive Representation of Objects with Edges**. Technical report, California Institute of Technology, 1999. 10

[18] Jean-Luc Starck, Emmanuel J. Candes, and David L. Donoho. **The Curvelet Transform for Image Denoising**. *IEEE Transactions on Image Processing*, **11**(6):670–684, 2002. 10

[19] Jean luc Starck, Fionn Murtagh, Emmanuel J. Cands, and David L. Donoho. **Gray and Color Image Contrast Enhancement by the Curvelet Transform**. *IEEE Transactions On Image Processing*, **12**(6), 2003. 10, 11

[20] Lexing Ying, Laurent Demanet, and Emmanuel C. **Fast discrete curvelet transforms**. *SIAM Multiscale Modeling Simulation*, (5):861–899, 2005. 10

[21] E. Candes, L. Demanet, D. Donoho, and L. Ying. **Fast discrete curvelet transforms**. *Multiscale Modeling Simulation*, **5**(3), 2006. 10

[22] Mohammad Saleh Miri and Ali Mahlooji Far. **Retinal Image Analysis Using Curvelet Transform and Multistructure Elements Morphology by Reconstruction**. *IEEE Trans. Biomed. Engineering*, **58**(5):1183–1192, 2011. 11

[23] Luc Vincent and Dominique Jeulin. **Minimal Paths and Crack Propagation Simulations**. In *Fifth European Congress for Stereology*, **8/2** of *Acta Stereologica*, pages 487–494, 1989. 11, 109

[24] Luc Vincent. **Minimal path algorithms for the robust detection of linear features in gray images**. In *Proceedings of the fourth International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, ISMM '98, pages 331–338, Norwell, MA, USA, 1998. Kluwer Academic Publishers. 11, 23, 109

[25] Vincent Bismuth, Régis Vaillant, Hugues Talbot, and Laurent Najman. **Curvilinear Structure Enhancement with the Polygonal Path Image - Application to Guide-Wire Segmentation in X-Ray Fluoroscopy**. In Nicholas Ayache, Hervé Delingette, Polina Golland, and Kensaku Mori, editors, *MICCAI (2)*, **7511** of *Lecture Notes in Computer Science*, pages 9–16. Springer, 2012. 11, 109

[26] Minsoo Suk and Ohyoung Song. **Curvilinear feature extraction using minimum spanning trees**. *Computer Vision, Graphics, and Image Processing*, **26**(3):400–411, 1984. 12

[27] Youssef Rouchdy and Laurent D. Cohen. **Image segmentation by geodesic voting. Application to the extraction of tree structures from confocal microscope images**. In *International Conference on Pattern Recognition*, pages 1–5, 2008. 12

[28] M. Carlotto. **Enhancement of Low-Contrast Curvilinear Features in Imagery**. *Transactions on Image Processing*, **16**(1):221–228, January 2007. 12

[29] Nagao M. and Matsuyama T. **Edge Preserving Smoothing**. *Computer Graphics and Image Processing*, **9**:394–407, 1979. 12

[30] Harwood D., Subbarao M., Hakalahti H., and Davis L. **A new class of edge-preserving smoothing filters**. *Pattern Recognition Letters*, **5**:155–162, 1987. 12

[31] Stephen M. Smith and J. Michael Brady. **SUSAN - A New Approach to Low Level Image Processing**. *International Journal of Computer Vision*, **23**(1):45–78, 1997. 13

[32] Carlo Tomasi and Roberto Manduchi. **Bilateral Filtering for Gray and Color Images**. In *International Conference on Computer Vision*, pages 839–846, 1998. 13

[33] Sylvain Paris and Frédo Durand. **A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach**. *International Journal of Computer Vision*, **81**(1):24–52, 2009. 13

[34] P. Perona and J. Malik. **Scale-Space and Edge Detection Using Anisotropic Diffusion**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(7):629–639, July 1990. 14

[35] Joachim Weickert. **A Review of Nonlinear Diffusion Filtering**. In *Proceedings of the First International Conference on Scale-Space Theory in Computer Vision*, SCALE-SPACE '97, pages 3–28, London, UK, UK, 1997. Springer-Verlag. 14

[36] Koichiro Deguchi, Tadahiro Izumitani, and Hidekata Hontani. **Detection and Enhancement of Line Structures in an Image by Anisotropic Diffusion**. In *Proceedings of the 4th International Workshop on Visual Form*, IWVF-4, pages 313–322, London, UK, 2001. Springer-Verlag. 14

[37] R. Manniesing, M.A. Viergever, and W.J. Niessen. **Vessel Enhancing Diffusion - A Scale Space Representation of Vessel Structures**. *Medical Image Analysis*, **10**:815–825, 2006. 14

[38] Francine Catté, Pierre-Louis Lions, Jean-Michel Morel, and Tomeu Coll. **Image selective smoothing and edge detection by nonlinear diffusion**. *SIAM J. Numer. Anal.*, **29**(1):182–193, February 1992. 14

[39] Karl Krissian, Grégoire Malandain, and Nicholas Ayache. **Directional Anisotropic Diffusion Applied to Segmentation of Vessels in 3D Images**. In *Proceedings of the First International Conference on Scale-Space Theory in Computer Vision*, SCALE-SPACE '97, pages 345–348, London, UK, UK, 1997. Springer-Verlag. 14

[40] Karl Krissian. **Flux-Based Anisotropic Diffusion Applied to Enhancement of 3-D Angiogram**. *IEEE Transactions on Medical Imaging*, **21**(11):1440–1442, 2002. 14

[41] M. Orkisz, C. Bresson, I.E. Magnin, O. Champin, and P. Douek. **Improved vessel visualization in MR angiography by non-linear anisotropic filtering**. *Magnetic Resonance in Medicine*, **37**:914–919, 1997. 14, 20

[42] R. Czerwinski, D. Jones, and Jr. O'Brien. **An approach to boundary detection in ultrasound imaging**. pages 951–955, 1993. 14

[43] Richard N. Czerwinski, Douglas L. Jones, and William D. O'brien. **Line and Boundary Detection in Speckle Images**. *IEEE Transactions on Image Processing*, **7**:1700–1714, 1997. 14

[44] R Kutka and S Stier. **Extraction of line properties based on direction fields.** *IEEE Transactions on Medical Imaging*, **15**(1):51–8, 1996. 14

[45] Haiguang Chen and James Hale. **An Algorithm for MR Angiography Image Enhancement**. *Magnetic Resonance in Medicine*, **33**(4):534–540, 1995. 14

[46] Y P Du, D L Parker, and W L Davis. **Vessel enhancement filtering in three-dimensional MR angiography.** *Journal of Magnetic Resonance Imaging*, **5**(2):151–7. 14

[47] Y. P. Du and D. L. Parker. **Vessel enhancement filtering in three-dimensional MR angiograms using long-range signal correlation**. *Journal of Magnetic Resonance Imaging*, **7**(2):447–450, 1997. 14

[48] Phan T. H. Truc, Md. A. U. Khan, Young-Koo Lee, Sungyoung Lee, and Tae-Seong Kim. **Vessel enhancement filter using directional filter bank**. *Computer Vision and Image Understanding*, **113**(1):101–112, January 2009. 14

[49] Jean Serra. *Image Analysis and Mathematical Morphology.* Academic Press, Inc., Orlando, FL, USA, 1983. 15, 109

[50] Jean Serra, editor. *Image Analysis and Mathematical Morphology. Volume 2. , Theoretical advances.* Academic Press, London, San Diego, New-York, 1988. 15, 109

[51] G. Matheron. *Random Sets and Integral Geometry.* Wiley, 1975. 15, 109

[52] Pierre Soille. *Morphological Image Analysis: Principles and Applications.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition, 2003. 15, 18, 23, 28, 109, 113

[53] Jean Serra and Luc Vincent. **An overview of morphological filtering**. In *Circuits, Systems and Signal Processing*, pages 47–108, 1992. 15, 109

[54] Laurent Najman and Hugues Talbot. *Morphologie Mathématique 1 : approches déterministes.* Hermès - Lavoisier, September 2008. (260 pp.). 15, 109

[55] Laurent Najman and Hugues Talbot. *Mathematical morphology: from theory to applications.* ISTE-Wiley, June 2010. ISBN: 9781848212152 (520 pp.). 15

[56] Benoít Naegel, Nicolas Passat, and Christian Ronse. **Grey-level hit-or-miss transforms-Part I: Unified theory**. *Pattern Recognition*, **40**(2):635–647, February 2007. 18

[57] Benoít Naegel, Nicolas Passat, and Christian Ronse. **Grey-level hit-or-miss transforms-part II: Application to angiographic image processing**. *Pattern Recognition*, **40**(2):648–658, February 2007. 18

[58] Bessem Bouraoui, Christian Ronse, Joseph Baruthio, Nicolas Passat, and Ph. Germain. **3D segmentation of coronary arteries based on advanced mathematical morphology techniques**. *Computerized Medical Imaging and Graphics*, **34**(5):377–387, 2010. 18

[59] Erchan Aptoula, Sébastien Lefèvre, and Christian Ronse. **A hit-or-miss transform for multivariate images**. *Pattern Recognition Letters*, **30**(8):760–764, 2009. 18

[60] Pierre Soille, Edmond J. Breen, and Ronald Jones. **Recursive Implementation of Erosions and Dilations Along Discrete Lines at Arbitrary Angles**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(5):562–567, 1996. 18, 110

[61] Jack Bresenham. **Algorithm for Computer Control of a Digital Plotter**. *IBM Systems Journal*, **4**(1):25–30, 1965. 18

[62] Pierre Soille and Hugues Talbot. **Directional Morphological Filtering**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(11):1313–1329, 2001. 18, 19, 34, 56

[63] Jocelyn Chanussot and Patrick Lanbert. **An application of mathematical morphology to road network extraction on SAR images**. In *Proceedings of the fourth International Symposium on Mathematical Morphology and its Applications to Image and Signal Processing*, ISMM '98, pages 399–406, Norwell, MA, USA, 1998. Kluwer Academic Publishers. 19

[64] A. Katartzis, V. Pizurica, and H. Sahli. **Application of Mathematical Morphology and Markov random field theory to the automatic extraction of linear features in airborne images**. In *In Mathematical Morphology and its Applications to Image and Signal Processing*, pages 405–414. Kluwer Academic Publishers, 2000. 19

[65] Pierre Soille and Hugues Talbot. **Image Structure Orientation Using Mathematical Morphology**. In *Proceedings of the 14th International Conference on Pattern Recognition-Volume 2 - Volume 2*, ICPR '98, pages 1467–, Washington, DC, USA, 1998. IEEE Computer Society. 19, 83

[66] Pierre Soille. **On morphological operators based on rank filters**. *Pattern Recognition*, **35**(2):527–535, 2002. 19, 34, 110

[67] Nidhal Bouaynaya, Mohammed Charif-Chefchaouni, and Dan Schonfeld. **Theoretical Foundations of Spatially-Variant Mathematical Morphology Part I: Binary Images**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **30**(5):823–836, 2008. 20

[68] Nidhal Bouaynaya and Dan Schonfeld. **Theoretical Foundations of Spatially-Variant Mathematical Morphology Part II: Gray-Level Images**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(5):837–850, 2008. 20

[69] O. Tankyevych, H. Talbot, and P. Dokládal. **Curvilinear morpho-Hessian filter**. In *IEEE ISBI*, pages 1011–1014, 2008. 20

[70] O. Tankyevych, H. Talbot, P. Dokládal, and N. Passat. **Direction-adaptive grey-level morphology. application to 3D vascular brain imaging**. In *IEEE ICIP*, pages 2261–2264, 2009. 20

[71] Olena Tankyevych, Hugues Talbot, Petr Dokládal, and Nicolas Passat. **Spatially-Variant Morpho-Hessian Filter: Efficient Implementation and Applications**. In *Ninth International Symposium on Mathematical Morphology*, Groningen, The Netherlands, 2009. 20, 104, 128

[72] O. Tankyevych. *Filtering of thin objects: applications to vascular image analysis*. PhD thesis, 2010. 20, 57

[73] R. Verd-Monedero, J. Angulo, and J. Serra. **Anisotropic Morphological Filters With Spatially-Variant Structuring Elements Based on Image-Dependent Gradient Fields**. *Transactions on Image Processing*, **20**(1):200–212, January 2011. 20

[74] Rafael Verdú-Monedero, Jesús Angulo, and Jean Serra. **Spatially-Variant Anisotropic Morphological Filters Driven by Gradient Fields**. In *Proceedings of the 9th International Symposium on Mathematical Morphology and Its Application to Signal and Image Processing*, ISMM '09, pages 115–125, Berlin, Heidelberg, 2009. Springer-Verlag. 20

[75] Michael H. F. Wilkinson and Michel A. Westenberg. **Shape Preserving Filament Enhancement Filtering**. In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, MICCAI '01, pages 770–777, London, UK, UK, 2001. Springer-Verlag. 20, 110

[76] Vincent Morard, Etienne Decencière, and Petr Dokládal. **Geodesic Attributes Thinnings and Thickenings**. In Pierre Soille, Martino Pesaresi, and Georgios K. Ouzounis, editors, *ISMM*, **6671** of *Lecture Notes in Computer Science*, pages 200–211. Springer, 2011. 20, 63, 69, 121, 124

[77] Vincent Morard, Etienne Decencière, and Petr Dokládal. **Efficient Geodesic Attribute Thinnings Based on the Barycentric Diameter**. *Journal of Mathematical Imaging and Vision*, pages 1–15, August 2012. 20, 63, 66, 69, 121, 124

[78] C. L. Luengo Hendriks, G. M. P. van Kempen, and L. J. van Vliet. **Improving the accuracy of isotropic granulometries**. *Pattern Recognition Letters*, **28**(7):865–872, May 2007. 23

[79] Cris L. Luengo Hendriks and Lucas J. Van Vliet. **A rotation-invariant morphology for shape analysis of anisotropic objects and structures**. In *In Proceedings 4th International Workshop on Visual Form, IWVF4, LNCS 2059*, pages 378–387. Springer, 2001. 23

[80] Cris L. Luengo Hendriks and Lucas J. van Vliet. **Using Line Segments as Structuring Elements for Sampling-Invariant Measurements**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(11):1826–1831, 2005. 23, 111

[81] Cris L. Luengo Hendriks. **Constrained and Dimensionality-Independent Path Openings**. *IEEE Transactions on Image Processing*, **19**(6):1587–1595, 2010. 23, 24, 29, 38, 43, 112, 113, 117

[82] Michael Buckley and Hugues Talbot. **Flexible linear openings and closings**. In John Goutsias, Luc Vincent, and Dan S. Bloomberg, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 109–118. Kluwer Academic Publishers, Dordrecht, 2000. 23, 34, 111

[83] Hugues Talbot and Ben Appleton. **Efficient complete and incomplete paths openings and closings**. *Image and Vision Computing*, **25**(4):416–425, 2007. 24, 28, 29, 34, 111, 113, 115

[84] Petros Maragos and Robert D. Ziff. **Threshold Superposition in Morphological Image Analysis Systems**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(5):498–504, 1990. 28, 113

[85] François Cokelaer, Hugues Talbot, and Jocelyn Chanussot. **Efficient Robust d-Dimensional Path Operators**. *Journal of Selected Topics in Signal Processing*, **6**(7):830–839, 2012. 38, 117

[86] Maysa M. G. Macedo, Choukri Mekkaoui, and Marcel Jackowski. **Vessel Centerline Tracking in CTA and MRA Images Using Hough Transform**. In Isabelle Bloch and Roberto M. Cesar, editors, *CIARP*, **6419** of *Lecture Notes in Computer Science*, pages 295–302. Springer, 2010. 45

[87] Alexandra Pacureanu, Chantal Revol-Muller, Jean-Loïc Rose, Maria Sanchez Ruiz, and Françoise Peyrin. **A vesselness-guided variational segmentation of cellular networks from 3D micro-CT**. In *ISBI*, pages 912–915. IEEE, 2010. 45

[88] Alexandra Pacureanu, Jerome Rollet, Chantal Revol-Muller, Vasile Buzuloiu, Max Langer, and Françoise Peyrin. **Segmentation of 3D cellular networks from SR-micro-CT images**. In *ISBI*, pages 1970–1973. IEEE, 2011. 45

[89] Michal Sofka and Charles V. Stewart. **Retinal Vessel Extraction Using Multiscale Matched Filters, Confidence and Edge Measures**. *IEEE Transactions on Medical Imaging*, **25**(12):1531–1546, December 2006. 47, 120

[90] Frédéric Zana and Jean-Claude Klein. **Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation**. *IEEE Transactions on Image Processing*, **10**(7):1010–1019, 2001. 47, 120

[91] J.J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, and B. van Ginneken. **Ridge based vessel segmentation in color images of the retina**. *IEEE Transactions on Medical Imaging*, **23**(4):501–509, 2004. 51, 53

[92] M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman. **Blood vessel segmentation methodologies in retinal images - A survey**. *Computer Methods and Program in Biomedicine*, **108**(1):407–433, October 2012. 51, 54

[93] M. D. Abramoff, P. J. Magelhaes, and S. J. Ram. **Image processing with ImageJ**. *Biophotonics Int*, **11**(7):36–42, 2004. 56

[94] C. E. Metz. **Basic principles of ROC analysis.** *Seminars in nuclear medicine*, **8**(4):283–298, October 1978. 56, 57

[95] C. Lantuejoul and F. Maisonneuve. **Geodesic methods in quantitative image analysis.** *Pattern Recognition*, **17**(2):177–187, 1984. 63, 67, 68, 123

[96] Edmond J. Breen and Ronald Jones. **Attribute Openings, Thinnings, and Granulometries**. *Computer Vision and Image Understanding*, **64**(3):377–389, 1996. 65, 66, 123

[97] Philippe Salembier, Albert Oliveras, Albert Oliveras Member, and Luis Garrido. **Anti-extensive Connected Operators for Image and Sequence Processing**, 1998. 66

[98] E. R. Urbach and M. H. F. Wilkinson. **Shape-Only Granulometries and Gray-Scale Shape Filters**. In *ISMM*, pages 305–314, 2002. 66

[99] Luc Vincent. **Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms**. *IEEE Transactions on Image Processing*, **2**:176–201, 1993. 72, 75, 124

[100] Maria Axelsson. **Estimating 3D fibre orientation in volume images**. In *International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. 83

[101] Maria Axelsson. **An Evaluation of Scale and Noise Sensitivity of Fibre Orientation Estimation in Volume Images**. In Pasquale Foggia, Carlo Sansone,

and Mario Vento, editors, *ICIAP*, **5716** of *Lecture Notes in Computer Science*, pages 975–984. Springer, 2009. 83

[102] Katharina Robb, Oliver Wirjadi, and Katja Schladitz. **Fiber Orientation Estimation from 3D Image Data: Practical Algorithms, Visualization, and Interpretation**. In Andreas König, Mario Köppen, Nikola Kasabov, and Ajith Abraham, editors, *HIS*, pages 320–325. IEEE Computer Society, 2007. 83

[103] O. Wirjadi, K. Schladitz, A. Rack, and T. Breuel. **Applications of anisotropic image filters for computing 2D and 3D-fiber orientations**. *Stereology and Image Analysis–10th European Congress of ISS*, pages 107–112, 2009. 83

[104] Christianne Mulat, Marc Donias, Pierre Baylou, Grard Vignoles, and Christian Germain. **Optimal orientation estimators for detection of cylindrical objects**. *Signal, Image and Video Processing*, **2**:51–58, 2008. 84

[105] C. Mulat, M. Donias, P. Baylou, G.L. Vignoles, and C. Germain. **Axis detection of cylindrical objects in three-dimensional images**. *Journal of Electronic Imaging*, **17**(3):031108–031108, 2008. 84

[106] Hyunmi Yang and W. Brent Lindquist. **Three-dimensional image analysis of fibrous materials**. pages 275–282, 2000. 84

[107] Franz Pfeifer, Kastner Johann, and FRotraud. **Method for three-dimensional evaluation and visualization of the distribution of fibres in glass-fibre reinforced injection molded parts by micro-X-ray computed tomography**. In *Conference on Nondestructive Testing*, 2008. 84

[108] Petr Dokládal and Dominique Jeulin. **3-D Extraction of Fibres from Microtomographic Images of Fibre-Reinforced Composite Materials**. In Michael H. F. Wilkinson and Jos B. T. M. Roerdink, editors, *ISMM*, **5720** of *Lecture Notes in Computer Science*, pages 126–136. Springer, 2009. 84, 92

[109] Andrew M Stein, David A Vader, Louise M Jawerth, David A Weitz, and Leonard M Sander. **An algorithm for extracting the network geometry of three-dimensional collagen gels.** *Journal of Microscopy*, **232**(3):463–75, 2008. 84

[110] William J. Schroeder, Lisa S. Avila, and William Hoffman. **Visualizing with VTK: A Tutorial**. *IEEE Computer Graphics and Applications*, **20**(5):20–27, September 2000. 84

[111] T. Y. Kong and A. Rosenfeld. **Digital topology: Introduction and survey**. *Computer Vision, Graphics, and Image Processing*, **48**(3):357–393, December 1989. 89, 90

[112] D.G. Morgenthaler and University of Maryland at College Park. Computer Science Center. *Three-dimensional Simple Points: Serial Erosion, Parallel Thinning, and Skeletonization*. Technical report: Computer Science Center. University of Maryland, 1981. 90

[113] Gilles Bertrand. **Simple points, topological numbers and geodesic neighborhoods in cubic grids**. *Pattern Recognition Letters*, **15**(10):1003–1011, October 1994. 90

[114] Gilles Bertrand and Grégoire Malandain. **A new characterization of three-dimensional simple points**. *Pattern Recognition Letters*, **15**(2):169–175, February 1994. 90

[115] Azriel Rosenfeld and John L. Pfaltz. **Sequential Operations in Digital Picture Processing**. *J. ACM*, **13**(4):471–494, October 1966. 91

[116] E.R. Davies and A.P.N. Plummer. **Thinning algorithms: A critique and a new methodology**. *Pattern Recognition*, **14**(16):53 – 63, 1981. 91

[117] Gilles Bertrand and Michel Couprie. **Transformations topologiques discrètes**. In Chassery Jean-Marc Coeurjolly David, Montanvert Annick, editor, *Géométrie discrète et images numériques*, pages 187–209. Hermès, 2007. incollection. 91

[118] Jeremy Siek, Lie-Quan Lee, and Andrew Lumsdaine. **Boost Graph Library**. http://www.boost.org/libs/graph/, June 2000. 93