2021/11/21 v0.6 IPOL article class

Published in Image Processing On Line on 2022–04–25.
Submitted on 2021–09–17, accepted on 2022–03–25.
ISSN 2105–1232 © 2022 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2022.397

Adaptive Anisotropic Morphological Filtering Based on Co-Circularity of Local Orientations

Samy Blusseau[1], Santiago Velasco-Forero[1], Jesús Angulo[1], Isabelle Bloch[2]

[1] Center for Mathematical Morphology, Mines Paris, PSL University, France
(samy.blusseau@minesparis.psl.eu, santiago.velasco@minesparis.psl.eu,
jesus.angulo@minesparis.psl.eu)
[2] Sorbonne Université, CNRS, LIP6, Paris, France
(isabelle.bloch@sorbonne-universite.fr)

Communicated by Bertrand Kerautret *Demo edited by* Bertrand Kerautret

Abstract

The algorithm presented in this paper is an application of a general framework for morphological processing of signals on weighted graphs. Here we apply it to images by defining what we call a co-circularity graph. In this graph, the vertices are the pixels and the weighted edges depend on a consistency criterion (co-circularity) between local orientations estimated from the structure tensors. This graph induces anisotropic adaptive morphological operators which are related both to anisotropic diffusion in images and path optimality in graphs. We present several applications such as the enhancement of fiber-like structures, completion of interrupted edges and the regularization of grayscale images. We also discuss the parameters setting depending on the application.

Source Code

The reviewed source code and documentation for this algorithm are available from the web page of this article[1]. Compilation and usage instructions are included in the `README.txt` file of the archive.

Keywords: mathematical morphology; adaptive filtering; tropical algebra; structure tensors

[1] https://doi.org/10.5201/ipol.2022.397

1 Introduction

Mathematical morphology (MM) first arose as a branch of non-linear image processing [25] and since then has been extended to a general mathematical framework based on lattice theory [2, 15, 22]. Its initial developments essentially concerned translation invariant operators, represented by kernels called *structuring elements*.

The need to analyze and filter shapes without corrupting them naturally gave rise to the so called *adaptive* or *spatially variant* mathematical morphology [6, 7, 15, 26], in which structuring elements could vary in space so as to adapt to the local structures of the images being processed. This framework was applied to design edge-preserving filters in general [9, 10, 18, 33], but also showed to be particularly relevant to the processing of thin anisotropic objects in images [29, 32], such as vessels, fibers or roads in satellite images.

To specifically address the morphological filtering of thin elongated shapes, an alternative to adaptive MM has been developed, known as the class of *path-based* morphological operators [14]. Path openings and closings consist in defining a graph on the pixels of an image and selecting structures composed of pixels forming a path of sufficient length in the graph. This idea inspired many subsequent refinements until recently [20, 21], when ranking the path openings responses in several adjacency directions allowed to efficiently discriminate between truly anisotropic shapes and blob-like shapes.

The method proposed in the present paper is at the intersection of these two classes of morphological approaches: adaptive mathematical morphology and path-based morphological operators. It is an extension of previous work [3], where morphological operators for signal processing on graphs were presented in a tropical algebraic framework. On the one hand, it relies on the definition of space-varying structuring elements. On the other hand, these structuring elements are also local adjacencies inducing a whole graph on the pixels of the images, and the iterated morphological operators can be easily interpreted in terms of paths, thanks to the tropical formulation. Compared to existing adaptive morphological methods, this one offers an easy to handle formulation and a path interpretation, as well as a generalization to non-flat structuring elements (that is, the adjacency relationship is not necessarily binary but can be weighted). Compared to existing path-based methods, which always define a priori the allowed sets of binary adjacencies, our approach adapts the graph definition to each image without being slower.

In [3] it was already shown that, when applied to a specific kind of graphs defined on images, here called co-circularity graphs, the proposed framework could be suited for adaptive morphological filtering and enhancement of elongated structures in images. In particular, processing the *anisotropy image* deduced from structure tensors is a new and original way to localize these structures. Examples of edge-preserving smoothing of images were also provided. The present paper aims at explaining in all details this method based on co-circularity graphs, in order to ensure its reproducibility and make it easier for users to handle. In Section 2 we recall important basic definitions of Mathematical Morphology and present our general framework for morphological processing of signals on graphs. Applied to co-circularity graphs, as defined in Section 3, it yields an efficient morphological algorithmic pipeline that we test and discuss in Section 4, before concluding. The corresponding pseudo-code is provided in Appendix A.

2 Morphological Operators for Signals on Graphs

2.1 Mathematical Morphology on Complete Lattices

We propose to briefly recall the algebraic basis of mathematical morphology in complete lattices.

Definition 1 (Complete lattice)**.** *A complete lattice is a partially ordered set $(\mathcal{L}, \leq)$ for which any subset $A \subseteq \mathcal{L}$ has a least upper bound, also called supremum and noted $\bigvee A$, and a greatest lower bound, also called infimum and noted $\bigwedge A$.*

Complete lattices are stable under two combinations: by supremum and by infimum. Hence, two classes of morphisms naturally emerge: those commuting with the supremum and those commuting with the infimum.

Definition 2 (Dilation, Erosion)**.** *Let $(\mathcal{L}, \leq)$ and $(\mathcal{L}', \leq')$ be two complete lattices. A dilation is a mapping $\delta : \mathcal{L} \to \mathcal{L}'$ such that for any index set I and any family $(x_i)_{i \in I}$,*

$$\delta(\bigvee_{i \in I} x_i) = \bigvee'_{i \in I} \delta(x_i).$$

Similarly, an erosion is a mapping $\varepsilon : \mathcal{L}' \to \mathcal{L}$ such that for any family $(x_i)_{i \in I}$

$$\varepsilon(\bigwedge'_{i \in I} x_i) = \bigwedge_{i \in I} \varepsilon(x_i).$$

In these expressions, $\bigvee$ and $\bigwedge$ (respectively $\bigvee'$ and $\bigwedge'$) denote the supremum and infimum according to $\leq$ (respectively $\leq'$).

It follows from the definition that dilations and erosions are increasing mappings. Dilations and erosions are strongly related as they are adjoint to each other. More precisely, each dilation $\delta : \mathcal{L} \to \mathcal{L}'$ has a unique *adjoint* erosion $\varepsilon : \mathcal{L}' \to \mathcal{L}$ defined by

$$\forall y \in \mathcal{L}', \varepsilon(y) = \bigvee \{x \in \mathcal{L}, \delta(x) \leq' y\}. \tag{1}$$

Conversely, to each erosion $\varepsilon : \mathcal{L}' \to \mathcal{L}$ corresponds a unique adjoint dilation $\delta : \mathcal{L} \to \mathcal{L}'$ defined by

$$\forall x \in \mathcal{L}, \delta(x) = \bigwedge' \{y \in \mathcal{L}', x \leq \varepsilon(y)\}. \tag{2}$$

Such a pair of erosion and dilation forms an *adjunction*, which is characterized by the following definition.

Definition 3 (Adjunction)**.** *Let $(\mathcal{L}, \leq)$ and $(\mathcal{L}', \leq')$ be two complete lattices, $\delta : \mathcal{L} \to \mathcal{L}'$ and $\varepsilon : \mathcal{L}' \to \mathcal{L}$ two mappings. The couple (ε, δ) is an adjunction if and only if for any $x \in \mathcal{L}$ and $y \in \mathcal{L}'$,*

$$\delta(x) \leq' y \iff x \leq \varepsilon(y).$$

Then we have the following result: if (ε, δ) is an adjunction, then δ is a dilation, ε is an erosion, and they verify Equations (1) and (2). The notion of adjunction also helps to show that dilations and erosions fix respectively the smallest and largest elements of lattices, that is: $\delta(\bigwedge \mathcal{L}) = \bigwedge' \mathcal{L}'$ and $\varepsilon(\bigvee' \mathcal{L}') = \bigvee \mathcal{L}$.

Two other classes of lattice operators are very important in mathematical morphology: openings and closings. They can be seen as non-linear projectors.

Definition 4 (Opening, Closing)**.** *Let $\mathcal{L}$ be a complete lattice. A mapping $\gamma : \mathcal{L} \to \mathcal{L}$ is an opening if and only if it is increasing, anti-extensive and idempotent, that is, for any $x, y \in \mathcal{L}$*

- $x \leq y \Rightarrow \gamma(x) \leq \gamma(y)$,
- $\gamma(x) \leq x$,

- $\gamma \circ \gamma(x) = \gamma(x)$.

Similarly, a mapping $\varphi : \mathcal{L} \to \mathcal{L}$ is a closing if and only if it is increasing, extensive and idempotent (only the second property changes, that is $x \leq \varphi(x)$).

Hence, an opening γ projects the elements of $\mathcal{L}$ onto its set of invariants $\gamma(\mathcal{L}) = \{\gamma(x), x \in \mathcal{L}\} = \{x \in \mathcal{L}, \gamma(x) = x\}$. For each $x \in \mathcal{L}$, $\gamma(x)$ minimizes the "gap" between x and $\gamma(\mathcal{L})$ in the sense: $\gamma(x) = \bigvee\{y \in \gamma(\mathcal{L}), y \leq x\}$. Similarly, $\varphi(x) = \bigwedge\{y \in \varphi(\mathcal{L}), x \leq y\}$. Note that $\gamma(\mathcal{L})$ is closed under supremum and $\varphi(\mathcal{L})$ under infimum.

Another fundamental result based on the adjunction is the following: if (ε, δ) is an adjunction, then $\gamma = \delta \circ \varepsilon$ is an opening and $\varphi = \varepsilon \circ \delta$ is a closing. What is more, we have $\delta \circ \varepsilon \circ \delta = \delta$ and $\varepsilon \circ \delta \circ \varepsilon = \varepsilon$ which shows that the invariants of γ and φ are $\gamma(\mathcal{L}') = \delta(\mathcal{L}) = \{\delta(x), x \in \mathcal{L}\}$ and $\varphi(\mathcal{L}) = \varepsilon(\mathcal{L}') = \{\varepsilon(y), y \in \mathcal{L}'\}$. Furthermore, we shall remark that if a dilation δ and an erosion ε form an adjunction and $k \in \mathbb{N}$, then the iterated dilation $\delta^k = \delta \circ \cdots \circ \delta$ and the erosion $\varepsilon^k = \varepsilon \circ \cdots \circ \varepsilon$ form an adjunction as well, yielding a whole family of openings $\gamma_k = \delta^k \circ \varepsilon^k$ and closings $\varphi_k = \varepsilon^k \circ \delta^k$ (which are different from iterations of γ and φ, of no interest as these are idempotent). It is easy to show that the family of openings $(\gamma_k)_{k\geq 1}$ is decreasing (that is, $i \leq j \Rightarrow \gamma_i \geq \gamma_j$), forming what is called a *granulometry*, while $(\varphi_k)_{k\geq 1}$ is increasing, forming an *anti-granulometry*. In the rest of the paper, we will often omit the composition symbol $\circ$ between morphological operators, and note for example $\gamma = \delta\varepsilon$ instead of $\delta \circ \varepsilon$.

Regarding iterated dilations and erosions, let us highlight two results that will matter in numerical computations. First, for any dilation δ, whenever $\delta^{k+1}(\mathbf{x}) = \delta^k(\mathbf{x})$ there is no point computing further iterations as we will get $\delta^{k+q}(\mathbf{x}) = \delta^k(\mathbf{x})$ for any $q \in \mathbb{N}$, and the same holds for the erosion. This is why in practice we will monitor changes after each iteration (see source code and Algorithms 2 and 3). Secondly, if in addition δ is extensive, like δ_W is, then its adjoint erosion ε is anti-extensive and we have $\varepsilon\delta^k(\mathbf{x}) \leq \delta^k(\mathbf{x})$ on the one hand, and $\varepsilon\delta^k(\mathbf{x}) = \varepsilon\delta^{k+1}(\mathbf{x}) = \varepsilon\delta\delta^k(\mathbf{x}) = \varphi\delta^k(\mathbf{x}) \geq \delta^k(\mathbf{x})$ on the other hand. Therefore, $\varepsilon\delta^k(\mathbf{x}) = \delta^k(\mathbf{x})$: it is not worth computing erosions of $\delta^k(\mathbf{x})$ either, and in particular $\varphi_k(\mathbf{x}) = \varepsilon^k\delta^k(\mathbf{x}) = \delta^k(\mathbf{x})$. Obviously, we have the analogous result $\gamma_k(\mathbf{x}) = \delta^k\varepsilon^k(\mathbf{x}) = \varepsilon^k(\mathbf{x})$ whenever $\varepsilon^{k+1}(\mathbf{x}) = \varepsilon^k(\mathbf{x})$.

Openings and closings are only two examples (although the most used ones) of a broader set of complete lattice endomorphisms, called *morphological filters*. A morphological filter on a complete lattice $\mathcal{L}$ is an *increasing and idempotent* mapping from $\mathcal{L}$ into itself. It has been shown that many morphological filters can be built by combining erosions and dilations, openings and closings [13, 26, 27]. A simple but useful kind of combinations yields the *Alternating sequential filters*.

Definition 5 (Alternating sequential filters (ASF))**.** *Let $(\xi_i)_{i\in\mathbb{N}^*}$ an increasing family of filters and $(\psi_i)_{i\in\mathbb{N}^*}$ a decreasing family of filters, such that $\psi_1 \leq \xi_1$.*

Then for any $i \geq 1$, the operators

$$M_i = \psi_i\xi_i \ldots \psi_2\xi_2\psi_1\xi_1 \qquad N_i = \xi_i\psi_i \ldots \xi_2\psi_2\xi_1\psi_1$$

are filters, called Alternated sequential filters.

In particular, this applies to the decreasing family of openings $(\gamma_k)_{k\geq 1}$ and increasing family of closings $(\varphi_k)_{k\geq 1}$ defined earlier, since $\gamma_1 \leq \varphi_1$. Such ASF are known to be particularly suited for image denoising.

2.2 Discrete Max-Plus Algebra

Tropical algebra refers to the study of algebras where the sum operation is replaced by an idempotent operator, namely max or min, and the product may be kept ("max/min-times algebras") or replaced

by the sum ("max/min-plus algebras"). Here we will focus on the max-plus algebra before presenting its links to weighted graphs and mathematical morphology.

We consider the set $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$ equipped with the following two operations, for any $x, y \in \mathbb{R}_{\max}$:

- "$\oplus$": $x \oplus y = \max(x, y)$
- "$\otimes$": $x \otimes y = x + y$

They are noted as special kinds of addition and multiplication for the properties they share with the usual arithmetical operations: both are commutative and associative; $\otimes$ is distributive over $\oplus$; operation $\oplus$ has a neutral element which is $-\infty$ and is a null element for $\otimes$; the neutral element for $\otimes$ is 0. However one specificity of $(\mathbb{R}_{\max}, \oplus, \otimes)$ is the idempotency of $\oplus$ ($\max(x, x) = x$), for which it is called an idempotent semifield [19][2].

The set of matrices with elements in $\mathbb{R}_{\max}$ also has its natural "max-plus" counterpart of linear operations, based on $\oplus$ and $\otimes$. Let A and B be two $m \times n$ and $n \times p$ matrices, respectively ($m, n, p \in \mathbb{N}^*$). Then $A \otimes B$ is the $m \times p$ matrix defined by $(A \otimes B)_{ij} = \oplus_{k=1}^{n} a_{ik} \otimes b_{kj} = \max_{1 \leq k \leq n}(a_{ik} + b_{kj})$. For two matrices of the same size we also have the $\oplus$ operation defined by $(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij})$. As for $(\mathbb{R}_{\max}, \oplus, \otimes)$, $\otimes$ and $\oplus$ are associative and $\otimes$ is distributive over $\oplus$. However here $\otimes$ is non commutative and we do not have invertibility for $\otimes$ of any "non zero" matrix, i.e. every matrix whose coefficients are not all $-\infty$.

In the remaining of the paper we will note A^k the k-th power of a square matrix A in the max-plus sense, for any integer $k \geq 0$ (for $k = 0$, A^k is the max-plus identity matrix, for which $a_{ii} = 0$ and $a_{ij} = -\infty$ for $i \neq j$). The coefficients of the matrix A^k will be noted $a_{ij}^{(k)}$, and n denotes a positive integer. Furthermore, for $n \in \mathbb{N}^*$ we will denote by $Mat_n(\mathbb{R}_{\max})$ the set of $n \times n$ square matrices with coefficients in $\mathbb{R}_{\max}$. The following definition sets the focus on a particular subset of $Mat_n(\mathbb{R}_{\max})$, that will be of particular interest to define morphological operators.

Definition 6 (Conservative morphological weight (CMW) matrix [3, 31])**.** *A matrix $W \in Mat_n(\mathbb{R}_{\max})$ is a* conservative morphological weight matrix *if $-\infty \leq w_{ij} \leq 0$ and $w_{ii} = 0$ for any $(i, j), 1 \leq i, j \leq n$.*

This class of matrices is a subset of the *doubly-0-astic* matrices [8], which are matrices with non-positive coefficients and at least one zero on each column and one on each row. We rather consider conservative morphological weight matrices as they represent extensive morphological dilations and anti-extensive erosions, which are more suited to our applications (see Section 4). In the rest of the paper we will shorten the denomination in *CMW matrices*. Note that the max-plus product of CMW matrices is a CMW matrix.

2.3 Operators on Weighted Graphs

Graphs with conservative morphological weights. Let $n \in \mathbb{N}^*$, $W = (w_{ij})_{1 \leq i,j \leq n}$ be a $n \times n$ CMW matrix (see Definition 6) and let $\mathcal{G}(W) = (V, E)$ be a weighted and directed graph containing n vertices whose adjacency matrix is W. As the w_{ij} belong to $\mathbb{R}_{\max}^- = \mathbb{R}^- \cup \{-\infty\}$, here the convention is that $w_{ij} > -\infty$ if and only if $(i, j) \in E$. We now recall and adapt a few definitions from graph theory (some can be found in [8]).

A *path* from vertex i to vertex j in $\mathcal{G}$ is a tuple of vertices $\gamma = (q_1, \ldots, q_l)$ such that $q_1 = i$, $q_l = j$, and $(q_k, q_{k+1}) \in E$ for $1 \leq k \leq l - 1$. Then the *length* of the path, noted $\ell(\gamma)$, is $l - 1$ (the number of its edges). For $k \geq 1$, $\Gamma_{ij}^{(k)}$ denotes the set of paths from i to j in $\mathcal{G}$ of length k. We note Γ_{ij}^{*} the set

[2]This structure is also called a *commutative belt with identity* in [8].

of paths from i to j in $\mathcal{G}$. Given the weight matrix W, the *weight of a path* $\gamma = (q_1, \ldots, q_l)$, noted $\omega(\gamma)$, is the sum $\omega(\gamma) = \sum_{k=1}^{l-1} w_{q_k q_{k+1}}$.

We will call *optimal path*, in a set of paths, a path that achieves the maximum weight among this set, when the maximum exists. In particular, an optimal path from a vertex i to a vertex j is any path γ^* such that $\omega(\gamma^*) = \max\{\omega(\gamma), \gamma \in \Gamma^*_{ij}\}$. In our case the latter maximum exists as there is a finite number of paths without circuit from a vertex to another and, since weights are non-positive, paths with circuits are necessarily suboptimal.

The following results show some strong links between tropical algebra and graphs. They can be found in various forms in the literature (e.g. [1, 5]) hence the proofs are omitted here. We recall that the powers W^k are intended in the max-plus sense (Section 2.2).

Proposition 1. *Let $W \in Mat_n(\mathbb{R}_{\max})$ and $k \in \mathbb{N}^*$. Then for any $1 \leq i, j \leq n$,*
1. $w_{ij}^{(k)} > -\infty$ if and only if there is at least a path in $\mathcal{G}(W)$ from vertex i to vertex j of length k;
2. $w_{ij}^{(k)}$ is the maximal weight of the set of paths from vertex i to vertex j of length k.

In short, Proposition 1 can be simply written

$$w_{ij}^{(k)} = \max_{\gamma \in \Gamma_{ij}^{(k)}} \omega(\gamma), \tag{3}$$

with the convention $\max(\emptyset) = -\infty$.

As said earlier, if W is a CMW matrix, then so is W^k. Its corresponding graph $\mathcal{G}(W^k)$ has the same set of vertices as the original graph $\mathcal{G}(W)$, but an edge exists between vertices i and j in $\mathcal{G}(W^k)$ whenever there is a path of length k from i to j in $\mathcal{G}(W)$. The weight associated with this new edge is $w_{ij}^{(k)} = \omega^*(\Gamma_{ij}^{(k)})$, the maximal weight for the paths from i to j of length k in $\mathcal{G}(W)$.

Morphological operators. In this section we link the max-plus product $W \otimes \mathbf{x}$ (Section 2.2) to morphological operators on a vector $\mathbf{x}$ supported by a graph $\mathcal{G}(W)$ - meaning that each value x_i is supported by vertex i. This formulation generalizes many of the dilations and erosions defined on graphs and in image processing applications.

For a fixed integer $n \geq 1$, let $0 \leq a < b \in \mathbb{R}^+$ be two non-negative real numbers and $\mathcal{L} = ([a, b]^n, \leq)$ be the complete lattice equipped with the usual product partial ordering (Pareto ordering): $\mathbf{x} \leq \mathbf{y} \iff x_i \leq y_i \ \ \forall i \in \{1, \ldots, n\}$. The supremum and infimum on $\mathcal{L}$ are induced by the Pareto ordering: for a family $(\mathbf{x}^{(k)})_{k \in K}$ of $\mathcal{L}$, $\bigvee_{k \in K} \mathbf{x}^{(k)}$ is the vector $\mathbf{y}$ defined by $y_i = \bigvee_{k \in K} x_i^{(k)}$, where K is any index set.

We now introduce the morphological framework on $\mathcal{L}$, based on the max-plus algebra. Let $W \in Mat_n(\mathbb{R}_{\max})$ be a CMW matrix. Then the function δ_W defined by

$$\delta_W : \left\{ \begin{array}{ccc} \mathcal{L} & \to & \mathcal{L} \\ \mathbf{x} & \mapsto & W \otimes \mathbf{x}, \end{array} \right. \tag{4}$$

which can be detailed by

$$\forall i \in \{1, \ldots, n\}, \ \delta_W(\mathbf{x})_i = (W \otimes \mathbf{x})_i = \oplus_{j=1}^n w_{ij} \otimes x_j = \bigvee_{1 \leq j \leq n} w_{ij} + x_j, \tag{5}$$

is an extensive morphological dilation on $\mathcal{L}$. As such, δ_W has an adjoint erosion defined by Equation (1). To get an explicit expression, one can show that for any $\mathbf{x}, \mathbf{y} \in \mathcal{L}$

$$\delta_W(\mathbf{x}) \leq \mathbf{y} \iff \forall i \in \{1, \ldots, n\}, x_i \leq \bigwedge_{1 \leq j \leq n} y_j - w_{ji}.$$

It follows from this and from Equation (1) that the erosion ε_W adjoint to δ_W is defined for $\mathbf{y} \in \mathcal{L}$ by

$$\forall i \in \{1, \dots, n\}, \quad \varepsilon_W(\mathbf{y})_i = \bigwedge_{1 \le j \le n} y_j - w_{ji}, \tag{6}$$

which can also be written

$$\varepsilon_W(\mathbf{y}) = -[W^T \otimes (-\mathbf{y})]. \tag{7}$$

We have now the basic morphological operators to process a signal $\mathbf{x} \in \mathcal{L}$ supported by a graph $\mathcal{G}(W)$, with W a $n \times n$ CMW matrix. Letting $\mathcal{N}_i \doteq \Big\{ j \in \{1, \dots, n\}, (i,j) \in E \Big\}$ denote the set of neighbors of vertex i in $\mathcal{G}(W)$ and $\check{\mathcal{N}}_i \doteq \Big\{ j \in \{1, \dots, n\}, i \in \mathcal{N}_j \Big\}$, the dilation and erosion of Equations (4) and (7) (or equivalently Equations (5) and (6)) can be written for $1 \le i \le n$

$$\delta_W(\mathbf{x})_i = \bigvee_{j \in \mathcal{N}_i} \Big(x_j + w_{ij} \Big), \quad \varepsilon_W(\mathbf{x})_i = \bigwedge_{j \in \check{\mathcal{N}}_i} \Big(x_j - w_{ji} \Big). \tag{8}$$

At this point one should remember that the w_{ij} are non positive. The closer w_{ij} is to 0, the more likely vertex j is to contribute to the sup (resp. the inf) in the dilation (resp. erosion) of vertex i. Conversely, the closer w_{ij} is to $-\infty$, the more likely vertex j is to be irrelevant in these computations. In fact, the supremum can be taken over $\mathcal{N}_i$ in the dilation because $w_{ij} = -\infty$ if $j \notin \mathcal{N}_i$, and the same holds for the erosion.

Note that Equation (8) generalizes many adjunctions in image processing. For example, setting $w_{ij} = \log(\mathbb{1}_{\mathcal{B}_j}(i))$ with $\mathcal{B}_j$ a set representing a structuring element at pixel j, we get the usual flat dilation and erosion. If the family $(\mathcal{B}_j)_{1 \le j \le n}$ is not translation invariant, we get the so called adaptive morphology framework [6, 7, 15, 26]; if $\mathcal{B}_j$ is not a spatial neighborhood of j, we get the typical non-local operators [23, 30]. In general, the formulation of Equation (8) defines non-flat, adaptive and possibly non-local morphological operators, and the structuring function at each pixel j is contained in the j-th column of matrix W, that we will note $W_{\cdot j}$.

It is also worth noticing, especially for numerical computing purposes, that in a CMW matrix W, a coefficient $w_{ij} \le a - b$ is equivalent to $w_{ij} = -\infty$ in the sense that the corresponding morphological operators defined in Equations (5) and (6) are unchanged. Indeed, since for any $1 \le i, j \le n$ we have $w_{ii} = 0$ and $x_j \in [a, b]$, any term $x_j + w_{ij}$ is not greater than a if $w_{ij} \le a - b$, and therefore not greater than $x_i + w_{ii} = x_i$. Hence it cannot contribute to the supremum in Equation (5), just like if we had $w_{ij} = -\infty$. The same reasoning can be done for the erosion. This is why in practice we can consider that $j \in \mathcal{N}_i$ if and only if $w_{ij} > a - b$, as will be done in the source code provided with this paper.

Finally, from the adjunction $(\varepsilon_W, \delta_W)$, many other operators can be derived, as described in Section 2.1. In particular, we have the corresponding opening $\gamma_W = \delta_W \varepsilon_W$ and closing $\varphi_W = \varepsilon_W \delta_W$. Furthermore, as said in Section 2.1 the iterated dilation $\delta_W^k = \delta_W \circ \cdots \circ \delta_W$ and the erosion $\varepsilon_W^k = \varepsilon_W \circ \cdots \circ \varepsilon_W$ form an adjunction. The associativity of the max-plus product $\otimes$ yields the following result for these iterated dilation and erosion.

Proposition 2. *Let W be a CMW matrix, and $k \in \mathbb{N}$. Then $\delta_W^k = \delta_{(W^k)}$ and $\varepsilon_W^k = \varepsilon_{(W^k)}$.*

This also implies the semigroup property $\delta_{(W^k)} \delta_{(W^l)} = \delta_{(W^{k+l})}$, and an easy writing for δ_W^k and ε_W^k

$$\delta_W^k(x)_i = \max_{j \in \mathcal{N}_i^k} \Big(x_j + w_{ij}^{(k)} \Big), \quad \varepsilon_W^k(x)_i = \min_{j \in \check{\mathcal{N}}_i^k} \Big(x_j - w_{ji}^{(k)} \Big), \tag{9}$$

where $\mathcal{N}_i^k$ is the set of neighbors of vertex i in $\mathcal{G}(W^k)$ or, equivalently, the set of vertices in $\mathcal{G}(W)$ that can be reached from i through a path of length k; and as before, $\check{\mathcal{N}}_i^k = \Big\{ j \in \{1, \dots, n\}, i \in \mathcal{N}_j^k \Big\}$.

Because of Proposition 2, it becomes natural to note $\gamma_{W^k} = \delta_W^k \varepsilon_W^k$ and $\varphi_{W^k} = \varepsilon_W^k \delta_W^k$ the openings and closings based on the adjoint iterated dilations and erosions. We shall now draw a link between path openings and γ_{W^k} in the next paragraph.

Path interpretation of openings γ_{W^k}. We may first recall the definition of a path opening [14] on $\mathbf{x} \in \mathcal{L}$ induced by a graph $\mathcal{G}$ represented by W and for a path length $k \in \mathbb{N}^*$. For $i \in \{1, \dots, n\}$, let us note $\mathcal{P}_i^k$ the set of paths of length at least k and passing through i in $\mathcal{G}$. Then the path opening $\Pi_W^{(k)}$ for a path length k is defined by

$$\Pi_W^{(k)}(\mathbf{x})_i = \bigvee \{t \in [a,b], \exists p \in \mathcal{P}_i^k, \forall l \in p,\ x_l \geq t\}, \tag{10}$$

with the convention that $\vee\{\emptyset\} = a$. Therefore, for $t \in (a,b]$, $\Pi_W^{(k)}(\mathbf{x})_i$ is larger than t if and only if vertex i belongs to a path of length at least k such that $\mathbf{x}$ takes values greater than t along that path

$$\Pi_W^{(k)}(\mathbf{x})_i \geq t \iff \exists p \in \mathcal{P}_i^k, \text{ such that } \forall l \in p,\ x_l \geq t. \tag{11}$$

Let us now give a similar formulation for γ_{W^k}. We can focus on γ_W since what follows holds for any CMW matrix, so we can replace W by W^k later on. As pointed out in Section 2.1, for any $\mathbf{x} \in \mathcal{L}$, $\gamma_W(\mathbf{x})$ is the best under-approximation of $\mathbf{x}$ by an invariant of γ_W, that is to say by some dilated element. Indeed, as $\gamma_W = \delta_W \varepsilon_W$, recall that $\gamma_W(\mathcal{L}) = \delta_W(\mathcal{L})$. Formally,

$$\gamma_W(\mathbf{x}) = \bigvee \{\mathbf{y} \in \delta_W(\mathcal{L}), \mathbf{y} \leq \mathbf{x}\}. \tag{12}$$

This point of view will help in the path interpretation. First, $\gamma_W(\mathbf{x})$ is the dilation $\delta_W(\mathbf{u})$ of some element $\mathbf{u}$ (actually, we know $\mathbf{u} = \varepsilon_W(\mathbf{x})$), that is to say the max-plus product $W \otimes \mathbf{u}$, which in turn is the sup-combination of the columns of W vertically translated

$$W \otimes \mathbf{u} = \oplus_{j=1}^n u_j \otimes W_{\cdot j} = \bigvee_{1 \leq j \leq n} u_j + W_{\cdot j}. \tag{13}$$

Secondly, this sup-combination is as close to $\mathbf{x}$ as possible, but is smaller than $\mathbf{x}$. Recalling that each column $W_{\cdot j}$ represents the neighborhood $\check{\mathcal{N}}_j$, or equivalently the structuring element associated with pixel j, we get that $\gamma_W(\mathbf{x})$ is composed as follows: for each j, lift the structuring element $\check{\mathcal{N}}_j$ vertically as high as possible as long as it remains under $\mathbf{x}$. Then take the upper envelope of these lifted structuring elements.

In the case where W is binary, that is to say $w_{ij} = 0$ if $j \in \mathcal{N}_i$ and $-\infty$ otherwise, then the structuring elements $\check{\mathcal{N}}_j$ are flat, and $\gamma_W(\mathbf{x})$ is an under-approximation of $\mathbf{x}$ by a superposition of *flat* regions, corresponding to the lifted $\check{\mathcal{N}}_j$s. Therefore if $\gamma_W(\mathbf{x})_i \geq t$ for some vertex i and value $t \in [a,b]$, then there is a whole neighborhood $\check{\mathcal{N}}_j$ to which i belongs, for which this also holds: $\forall l \in \check{\mathcal{N}}_j,\ \ \gamma_W(\mathbf{x})_l \geq t$. Since $\mathbf{x} \geq \gamma_W(\mathbf{x})$, we also get that $\mathbf{x}$ must be larger than t over $\check{\mathcal{N}}_j$. As we will see in Proposition 3, the converse is also true, and we get the following characterization

$$\gamma_W(\mathbf{x})_i \geq t \iff \exists j \in \mathcal{N}_i, \text{ such that } \forall l \in \check{\mathcal{N}}_j\ \ x_l \geq t. \tag{14}$$

Now, replacing W by W^k, for some integer $k \geq 1$, on the left hand side of Equation (14) leads to change $\mathcal{N}_i$ and $\check{\mathcal{N}}_j$ for $\mathcal{N}_i^k$ and $\check{\mathcal{N}}_j^k$ on the right hand side. Hence $\gamma_{W^k}(\mathbf{x})_i \geq t$ if and only if there is a vertex $j \in \mathcal{N}_i^k$ such that for *all* paths of length k ending in j (including the one linking i to j) $\mathbf{x}$ is greater than t along these paths. We can write, **in the binary case**

$$\gamma_{W^k}(\mathbf{x})_i = \bigvee \{t \in [a,b], \exists j \in \mathcal{N}_i^k,\ \forall l \in \check{\mathcal{N}}_j^k,\ \ x_l \geq t\}. \tag{15}$$

This is theoretically stronger than a path opening of length k. However, here paths may contain cycles whereas usually path openings are computed in acyclic graphs. Note in particular that every vertex is its own neighbor as W is a CMW matrix, and therefore even if the longest acyclic path from i to j is shorter than k, we still have $i \in \check{\mathcal{N}}_j^k$. Yet in practice, like in the experiments of Section 4 using the graphs defined in Section 3, most vertices $l \in \check{\mathcal{N}}_j^k$, belong to an acyclic path of length k ending in j. This can be observed in results images like in Figures 8 and 9, where large flat regions appear, corresponding to a bundle of paths.

In the general case, where W is a CMW matrix but no longer binary valued, Equation (14) takes the following formulation, proven in Appendix B.

Proposition 3 (Path interpretation of γ_{W^k})**.**

$$\gamma_W(\mathbf{x})_i \geq t \iff \exists j \in \mathcal{N}_i, \quad \textit{such that } \forall l \in \check{\mathcal{N}}_j \;\; x_l \geq t - w_{ij} + w_{lj}. \tag{16}$$

We easily check that Proposition 3 applied to the binary case, where $w_{ij} = 0 = w_{lj}$ for $j \in \mathcal{N}_i$ and $l \in \check{\mathcal{N}}_j$, yields Equation (14). Again, we can replace W by W^k for some integer $k \geq 1$, and accordingly change $\mathcal{N}_i$ and $\check{\mathcal{N}}_j$ for $\mathcal{N}_i^k$ and $\check{\mathcal{N}}_j^k$. Compared to Equation (14), the additional terms $-w_{ij} + w_{lj}$ quantify the strength of the connections of vertices in $\check{\mathcal{N}}_j$ to vertex j. The stronger these connections (that is, the closer the weights are to zero) the tighter the constraint on the values of $\mathbf{x}$ over $\check{\mathcal{N}}_j$ and therefore the flatter $\mathbf{x}$ on $\check{\mathcal{N}}_j$. An example of comparison between openings $\gamma_W(\mathbf{x})$ in the binary and non-binary cases is given in Figure 9, where $\mathbf{x}$ is an anisotropy image (see Sections 3 and 4.1). We see on these examples that in the non-binary case, flat zones are replaced by quasi-flat zones. As corollary of Proposition 3, we conclude this section by the path formulation of γ_{W^k} **in the general case**

$$\gamma_{W^k}(\mathbf{x})_i = \bigvee \{t \in [a,b], \exists j \in \mathcal{N}_i^k, \; \forall l \in \check{\mathcal{N}}_j^k, \;\; x_l \geq t - w_{ij}^{(k)} + w_{lj}^{(k)}\}. \tag{17}$$

3 Co-Circularity Graphs on Grayscale Images

In this section we introduce a class of graphs on images, based on the co-circularity of the structure tensors principal directions. Then the morphological operators defined in Section 2.3 will be of particular interest for the detection and enhancement of elongated shapes, as demonstrated in Section 4.

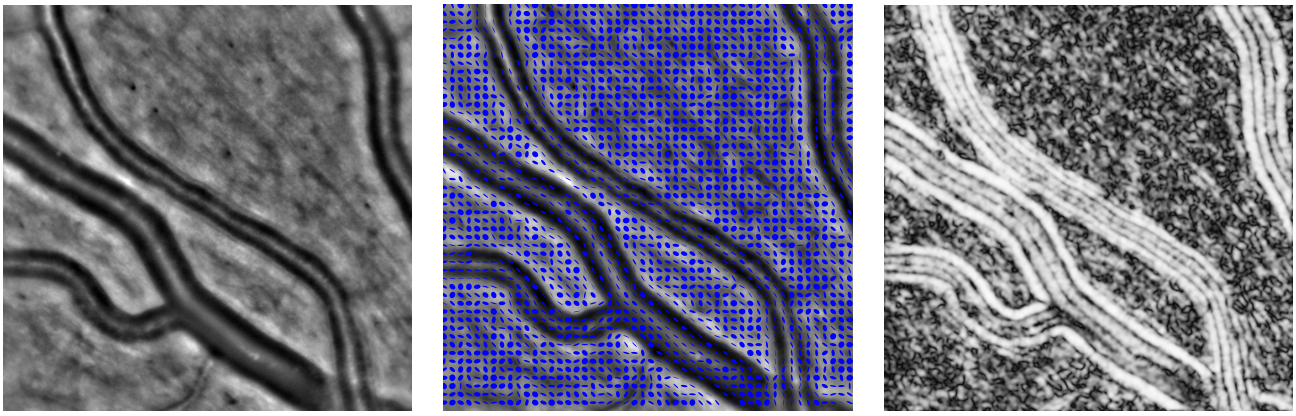
3.1 Structure Tensors

We use structure tensors [11, 12, 17] to estimate locally the main direction of anisotropic objects at different scales. Structure tensors can be seen as a second order local approximation of an image and as such they provide more than a local dominant direction: they also indicate the degree of anisotropy of the local structures.

Given an image f, the field T_0 of scale zero structure tensors maps each pixel (x, y) to the symmetric positive semidefinite matrix with rank ≤ 1,

$$T_0(x,y) = \nabla f(x,y) \cdot \nabla f(x,y)^T. \tag{18}$$

The gradient $\nabla f(x,y)$ is an eigenvector of $T_0(x,y)$ with $||\nabla f(x,y)||^2$ as corresponding eigenvalue, and 0 is the other eigenvalue. The eigenline associated with the smallest eigenvalue is therefore



(a) (b) (c)

Figure 1: (a) A grayscale image, (b) its structure tensors T_σ for $\sigma = 3$ pixels and (c) the corresponding anisotropy image. In (b) an ellipse is plotted every 15 pixels in both directions.

orthogonal to the gradient $\nabla f(x, y)$. Then for $\sigma > 0$ the structure tensor T_σ is the smoothed version

$$T_\sigma = G_\sigma * T_0, \tag{19}$$

where G_σ is the σ-scale Gaussian kernel. Note that, for any $u = (x, y)$, $T_\sigma(u)$ is a weighted sum of tensors $T_0(\cdot)$ with positive weights, and is therefore a positive semi-definite matrix that can be represented by the ellipse

$$\mathcal{E}_u^{(\sigma)} = \{v \in \mathbb{R}^2, v^T T_\sigma(u) v \leq 1\}, \tag{20}$$

(see Figure 1(b)). We note $0 \leq \lambda_2^{(\sigma)}(u) \leq \lambda_1^{(\sigma)}(u)$ the eigenvalues of $T_\sigma(u)$ and $(e_1^{(\sigma)}(u), e_2^{(\sigma)}(u))$ an associated basis of orthogonal eigenvectors. The second eigenvector $e_2^{(\sigma)}(u)$ indicates a direction associated to a low variation of image intensity at pixel u (it is roughly orthogonal to the gradient). It corresponds to the large axis of the ellipse representing $T_\sigma(u)$. We note

$$\theta^{(\sigma)}(u) = \arg(e_2^{(\sigma)}(u)). \tag{21}$$

From the tensor field T_σ we can build the anisotropy image

$$I_a^{(\sigma)} = \frac{\lambda_1^{(\sigma)} - \lambda_2^{(\sigma)}}{\lambda_1^{(\sigma)} + \lambda_2^{(\sigma)}}. \tag{22}$$

Then for any u, $0 \leq I_a^{(\sigma)}(u) \leq 1$, and the closer $I_a^{(\sigma)}(u)$ to 1, the more $T_\sigma(u)$ is anisotropic (see Figures 1 and 2). As illustrated by Figure 2, increasing σ smoothes the local estimation of anisotropy and main orientations.

3.2 Co-Circularity

Given two points $u_1, u_2 \in \mathbb{R}^2$ and two angles θ_1, θ_2 we say that (u_1, θ_1) and (u_2, θ_2) are co-circular if there is a circle (with possibly infinite radius) tangent in u_1 and u_2 to the lines directed by the vectors $[\cos(\theta_1), \sin(\theta_1)]^T$ and $[\cos(\theta_2), \sin(\theta_2)]^T$, respectively. Figure 3 (left) shows a set of vectors $u_i + [\cos(\theta_i), \sin(\theta_i)]^T, i \geq 1$ that are co-circular to the central horizontal vector $u_0 + [\cos(\theta_0), \sin(\theta_0)]^T$, with $u_0 = (0, 0)$ and $\theta_0 = 0$ rad. In practice, co-circularity between (u_1, θ_1) and (u_2, θ_2) can be tested as follows. Let $\mathbf{v_i} = [\cos(\theta_i), \sin(\theta_i)]^T$, $i = 1, 2$, $\mathbf{d} = \frac{u_2 - u_1}{||u_2 - u_1||_2}$ and $\mathbf{v_1'} = 2(\mathbf{v_1} \cdot \mathbf{d})\mathbf{d} - \mathbf{v_1}$ (that is, $\mathbf{v_1'}$ is simply the mirror symmetric of $\mathbf{v_1}$ with respect to the line directed by $\mathbf{d}$). Then (u_1, θ_1) and (u_2, θ_2)

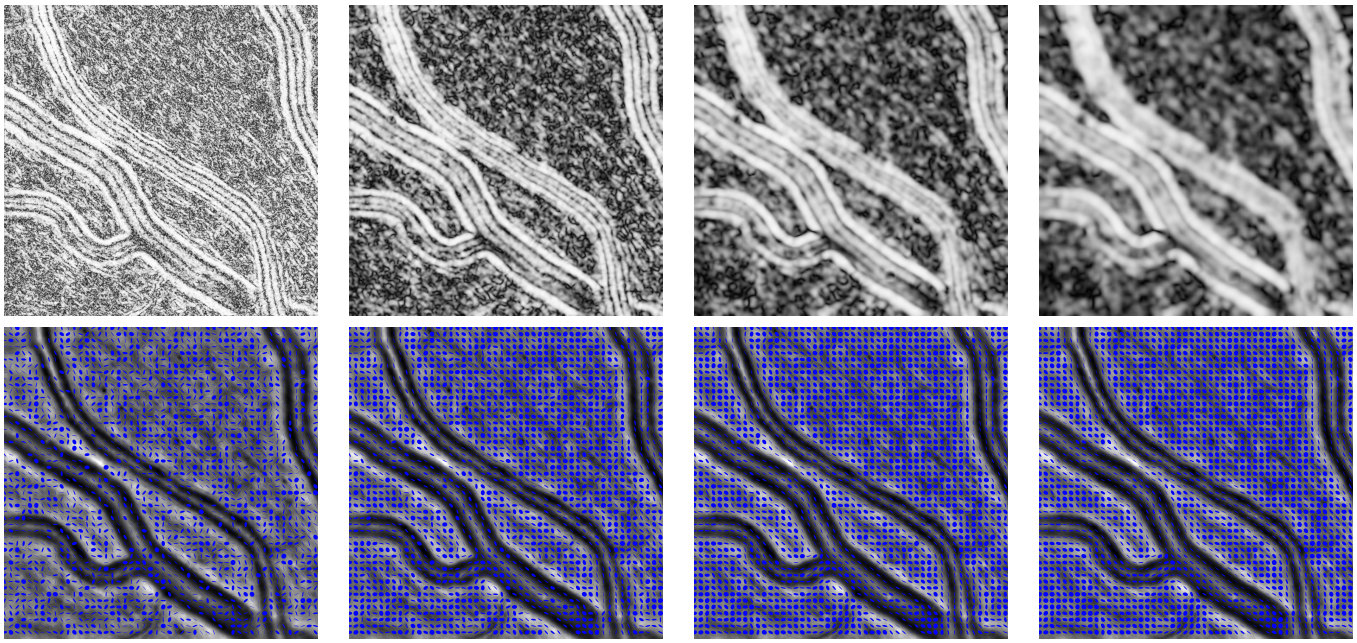


Figure 2: Anisotropy images (top row) associated with smoothed structure tensors T_σ (bottom row, subsampled every 15 pixels) for several values of σ. From left to right, $\sigma = 1, 3, 5, 7$ pixels. The grayscale input image is the one shown in Figure 1 (a).

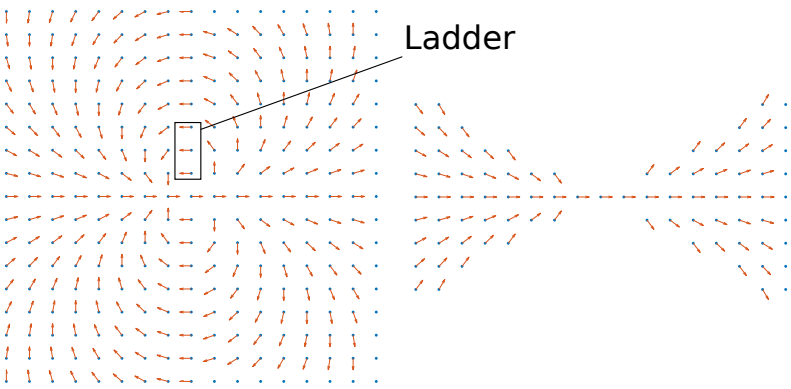


Figure 3: **Left:** a set of elements (u, θ) co-circular to the central element (u_0, θ_0) where u_0 is the origin and $\theta_0 = 0$ rad; **Right:** the elements from the left hand image that comply with the additional conic condition to avoid ladder configurations.

are co-circular if and only if $\mathbf{v_1'}$ and $\mathbf{v_2}$ are colinear, that is to say if and only if $|\mathbf{v_1'} \cdot \mathbf{v_2}| = 1$. This is illustrated in Figure 4. Note that the co-circularity condition does not depend on $||u_2 - u_1||$.

In order to deal with real data where perfect co-circularity almost never occurs, we define a relaxed condition up to an angular precision $\alpha \in [0, \frac{\pi}{2})$. The new condition becomes

$$(u_1, \theta_1) \text{ and } (u_2, \theta_2) \;\; \alpha\text{-co-circular} \iff |\mathbf{v_1'} \cdot \mathbf{v_2}| \geq \cos(\alpha), \tag{23}$$

as $|\mathbf{v_1'} \cdot \mathbf{v_2}|$ is the absolute value of the cosine of the angle between the unitary vectors $\mathbf{v_1'}$ and $\mathbf{v_2}$. From now on we will note

$$(u_1, \theta_1) \sim_\alpha (u_2, \theta_2)$$

when (u_1, θ_1) and (u_2, θ_2) are α-co-circular, as defined by Equation (23). Note that this relation is symmetric.

We use co-circularity as a mathematical criterion to model the Gestalt *good continuation* grouping law [16]. In other words, a list of couples $(u_i, \theta_i), i \geq 0$ such that (u_i, θ_i) is co-circular to (u_{i+1}, θ_{i+1})

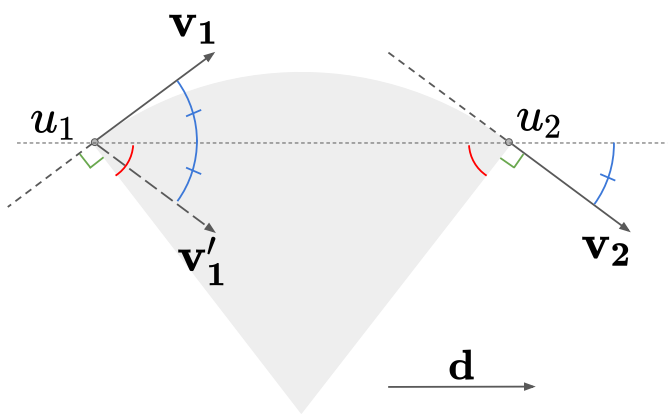


Figure 4: Configuration where $(u_1, \arg(\mathbf{v_1}))$ and $(u_2, \arg(\mathbf{v_2}))$ are co-circular. Angles with equal absolute values are represented with same colors and, where there is a possible ambiguity, they are marked by the same short segment.

is supposed to draw a (subsampled) smooth trajectory. As we can see on Figure 3 (left), the co-circularity criterion alone allows to chain elements in *ladder* configurations, which do not comply with the good continuation law. For this reason, when defining a graph on pixels based on local orientations, we will add what we call a *conic condition* on the adjacency of two vertices. Namely, for (u_2, θ_2) to be a neighbor of (u_1, θ_1) these couples should be co-circular *and* u_2 should lie within the double cone noted $\mathcal{C}(u_1, \theta_1, \beta)$ pointing at u_1, of axis the direction defined by θ_1 and of fixed solid angle $\beta \in [0, \frac{\pi}{2})$ (see Figure 3, right). Formally,

$$u_2 \in \mathcal{C}(u_1, \theta_1, \beta) \iff |\mathbf{d} \cdot \mathbf{v_1}| \geq \cos(\beta). \tag{24}$$

3.3 Co-Circularity Graphs

With the concepts of Sections 3.1 and 3.2, we can now define a graph on the pixels of an image, basing the adjacency relationship on co-circularity and structure tensors at a certain scale $\sigma > 0$.

Given a $\mathcal{W} \times \mathcal{H}$ grayscale image I with values in $[0, M]$, $M > 0$, and a scale $\sigma > 0$, we define a graph $\mathcal{G} = (V, E)$ containing $n = \mathcal{W} \cdot \mathcal{H}$ vertices, each vertex corresponding to one pixel. By convention, we associate the pixel of coordinates (r, c), $0 \leq r \leq \mathcal{W} - 1$, $0 \leq c \leq \mathcal{H} - 1$, to the vertex of index $r \times \mathcal{W} + c$. Therefore, to each vertex i, $0 \leq i \leq n - 1$, can be associated the attributes of the corresponding pixel, and in particular its coordinates $u_i = (r_i, c_i) = (\lfloor i/\mathcal{W} \rfloor, i - \lfloor i/\mathcal{W} \rfloor)$ as well as the main orientation $\theta_i^{(\sigma)} = \theta^{(\sigma)}(u_i)$ of the structure tensor $T_\sigma(u_i)$, as defined by Equation (21). The edges between vertices are defined based on these two attributes, as follows.

We represent $\mathcal{G}$ by its adjacency matrix $W = (w_{ij})_{1 \leq i,j \leq n}$, using the convention of Section 2.3: vertex j is neighbor of i if and only if $w_{ij} > -\infty$. Then we propose two alternatives to define the edges weights: the first one, where each weight w_{ij} can only take its value in $\{0, -\infty\}$, induces what we call a binary-weighted graph; the second one, where each weight $w_{ij} \in [-\infty, 0]$, induces a non binary-weighted graph.

Binary-weighted graph. In this case, the definition of the co-circularity graph requires four parameters:

- a scale $\sigma > 0$
- an angular precision $\alpha \in [0, \frac{\pi}{2})$ to relax the co-circularity constrain,
- an angle $\beta \in [0, \frac{\pi}{2})$ to define the conic condition,

- a positive integer p defining a $(2p+1) \times (2p+1)$ square neighborhood.

Then for two vertices i and j, $0 \leq i, j \leq n-1$, the edge weight w_{ij} is defined by

$$w_{ij} = \begin{cases} 0 & \text{if } (u_i, \theta_i^{(\sigma)}) \sim_\alpha (u_j, \theta_j^{(\sigma)}) \quad \text{and } u_j \in \mathcal{C}(u_i, \theta_i^{(\sigma)}, \beta) \cap \mathcal{S}(u_i, p), \\ -\infty & \textit{otherwise}, \end{cases} \tag{25}$$

where $\mathcal{S}(u_i, p)$ is the $(2p+1) \times (2p+1)$ square centered on u_i, and the α-co-circularity and the conic condition $u_j \in \mathcal{C}(u_i, \theta_i^{(\sigma)}, \beta)$ are defined as in Equations (23) and (24) respectively, replacing indices 1 and 2 by i and j. More precisely

$$\begin{aligned} \mathbf{v_i} &= [\cos(\theta_i^{(\sigma)}), \sin(\theta_i^{(\sigma)})]^T, & \mathbf{v_j} &= [\cos(\theta_j^{(\sigma)}), \sin(\theta_j^{(\sigma)})]^T, \\ \mathbf{d} &= \tfrac{u_j - u_i}{||u_j - u_i||_2}, & \mathbf{v_i'} &= 2(\mathbf{v_i} \cdot \mathbf{d})\mathbf{d} - \mathbf{v_i}, \\ (u_i, \theta_i^{(\sigma)}) \sim_\alpha (u_j, \theta_j^{(\sigma)}) &\iff & |\mathbf{v_i'} \cdot \mathbf{v_j}| &\geq \cos(\alpha), \\ u_j \in \mathcal{C}(u_i, \theta_i^{(\sigma)}, \beta) &\iff & |\mathbf{d} \cdot \mathbf{v_i}| &\geq \cos(\beta), \\ u_j \in \mathcal{S}(u_i, p) &\iff & ||u_i - u_j||_\infty &\leq p. \end{aligned} \tag{26}$$

Non binary-weighted graph. This alternative is meant to avoid the parameter α. Instead of setting a minimum threshold on the co-circularity precision, we ensure that w_{ij} decreases as this precision decreases. Formally,

$$w_{ij} = \begin{cases} M \cdot \log(|\mathbf{v_i'} \cdot \mathbf{v_j}|) & \text{if } u_j \in \mathcal{C}(u_i, \theta_i^{(\sigma)}, \beta) \cap \mathcal{S}(u_i, p), \\ -\infty & \textit{otherwise}, \end{cases} \tag{27}$$

using the definitions of Equations (26).

Remarks on the symmetry of W. Note that in the binary case and for $\alpha = 0$ (perfect co-circularity), the adjacency relationship is symmetric, that is: $w_{ij} = 0 \iff w_{ji} = 0$, and therefore $w_{ij} = w_{ji}$. However, for $\alpha > 0$ this is not the case anymore. Indeed, although the α-co-circularity remains symmetric, α-co-circular configurations exist where the conic condition is fulfilled in one direction only, as shown in Figure 5. For the same reason, in the non-binary case $w_{ij} \neq w_{ji}$ as soon as $u_j \in \mathcal{C}(u_i, \theta_i^{(\sigma)}, \beta)$ and $u_i \notin \mathcal{C}(u_j, \theta_j^{(\sigma)}, \beta)$, or vice versa (for two pixels i and j which are close enough, that is $||u_i - u_j||_\infty \leq p$).

In the binary case and for $\alpha \leq \beta$ (which is a natural choice) we can show that the probability of $w_{ij} \neq w_{ji}$ for random, independent and uniformly distributed $\theta_i^{(\sigma)}$ and $\theta_j^{(\sigma)}$, is low (typically, lower than 0.01 for $\alpha \leq \frac{\pi}{20}$). In practice, where $\theta_i^{(\sigma)}$ and $\theta_j^{(\sigma)}$ are correlated (even more so as σ increases), the observed frequency of $w_{ij} \neq w_{ji}$ is even lower, by an order of magnitude at least.

In the non-binary case, $w_{ij} \neq w_{ji}$ might occur more often but in practice when one of the weight equals $-\infty$, then its symmetric also has a low value and is unlikely to contribute to the morphological operators (which is our final aim).

Therefore, in Algorithm 1, and more precisely in the `ComputeAdjacencyMatrix` function we always assume the adjacency matrix to be symmetric, which allows to test adjacency between pairs of pixels only once. In practice, this means that when browsing the image pixel by pixel, we need not examine the whole $(2p+1) \times (2p+1)$ square neighborhood $\mathcal{S}(u_i, p)$ around each pixel i. Instead, we can restrict ourselves to examine only a subregion $\mathcal{R}(u_i, p)$ covering half of that square - and

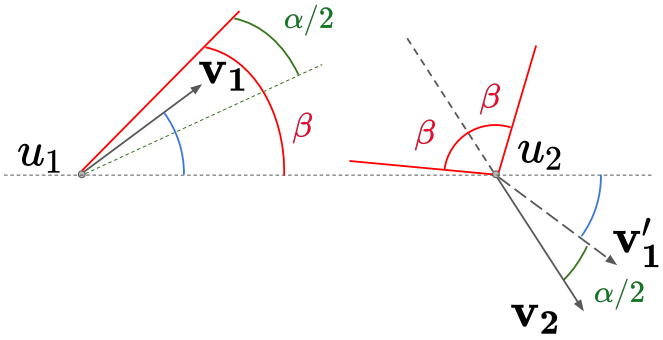


Figure 5: An example where $w_{12} = 0$ whereas $w_{21} = -\infty$. Indeed, $(u_1, \arg(\mathbf{v_1})) \sim_\alpha (u_2, \arg(\mathbf{v_2}))$ since $|\mathbf{v'_1} \cdot \mathbf{v_2}| = \cos(\frac{\alpha}{2}) \geq \cos(\alpha)$, and $u_2 \in \mathcal{C}(u_1, \arg(\mathbf{v_1}), \beta)$ but $u_1 \notin \mathcal{C}(u_2, \arg(\mathbf{v_2}), \beta)$.

excluding u_i, as we already know that i is its own neighbor in the graph ($w_{ii} = 0$). Formally, for each pair of pixels $i, j \in \{0, \ldots, n-1\}$, $j \neq i$, with coordinates $u_i = (r_i, c_i)$ and $u_j = (r_j, c_j)$, the definition of the subregion $\mathcal{R}$ must fulfil the following condition

$$||u_i - u_j||_\infty \leq p \iff \begin{cases} u_j \in \mathcal{R}(u_i, p) \\ u_i \notin \mathcal{R}(u_j, p) \end{cases} \text{or} \begin{cases} u_i \in \mathcal{R}(u_j, p) \\ u_j \notin \mathcal{R}(u_i, p) \end{cases}, \tag{28}$$

which means that testing the adjacency for every u_i and $u_j \in \mathcal{R}(u_i, p)$ ensures that every pair such that $||u_i - u_j||_\infty \leq p$ is tested once and only once. Many definitions are possible for $\mathcal{R}(u_i, p)$, and they are all independent from the way we browse the pixels of the image. Here we choose to define

$$\mathcal{R}(u_i, p) = \Big([r_i - 1, r_i + p] \times [c_i, c_i + p]\Big) \cup \Big([r_i - p, r_i] \times [c_i + 1, c_i + p]\Big), \tag{29}$$

as illustrated on Figure 6. From this drawing, one can check that condition (28) is fulfilled.

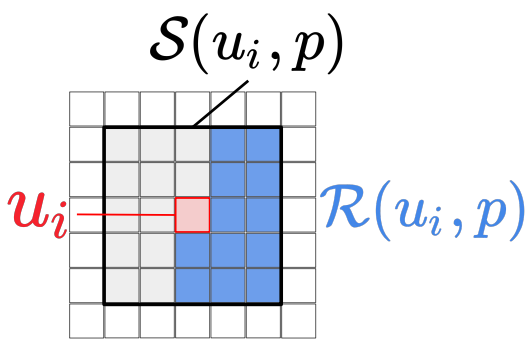


Figure 6: Illustration of the region $\mathcal{R}(u_i, p) \subset \mathcal{S}(u_i, p)$ for a pixel position u_i and a size parameter $p = 2$.

Remarks on the extension to higher dimensions. Co-circularity graphs based on structure tensors can be generalized to scalar data defined on n-dimensional grids, and in particular in three dimensions. Indeed, zero-scale structure tensors can be computed from the n-dimensional gradient and regularized with an isotropic Gaussian blurring in n dimensions. Then the main direction $\mathbf{v_i}$ is determined by the normalized eigenvector associated with the smallest eigenvalue (without the necessity to define an angle) just like in two dimensions, and the anisotropy image can be defined as in Equation (22), replacing $\lambda_2^{(\sigma)}$ and $\lambda_1^{(\sigma)}$ by the smallest and second smallest eigenvalues ($\lambda_n^{(\sigma)}$ and $\lambda_{n-1}^{(\sigma)}$). Finally, the co-circularity can also be defined through scalar products as in Equation (26),

except that an additional condition is necessary to test that $\mathbf{d}$ is in the plane defined by $\mathbf{v_i}$ and $\mathbf{v_j}$.

In this section we have built co-circularity graphs represented by CMW adjacency matrices. In the next one we show the results of applying the corresponding morphological operators, as defined in Section 2 by Equation (8), to grayscale images.

4 Applications

Before showing experimental results, let us detail in Algorithms 2 and 3 (see Appendix A) the implementation of the morphological operators of Equation (8), in the setting of Section 3 where the matrix W is sparse (and symmetric). Algorithms 1, 2 and 3 are the fundamental bricks for all the morphological processing presented hereafter.

4.1 Processing of Anisotropy Images

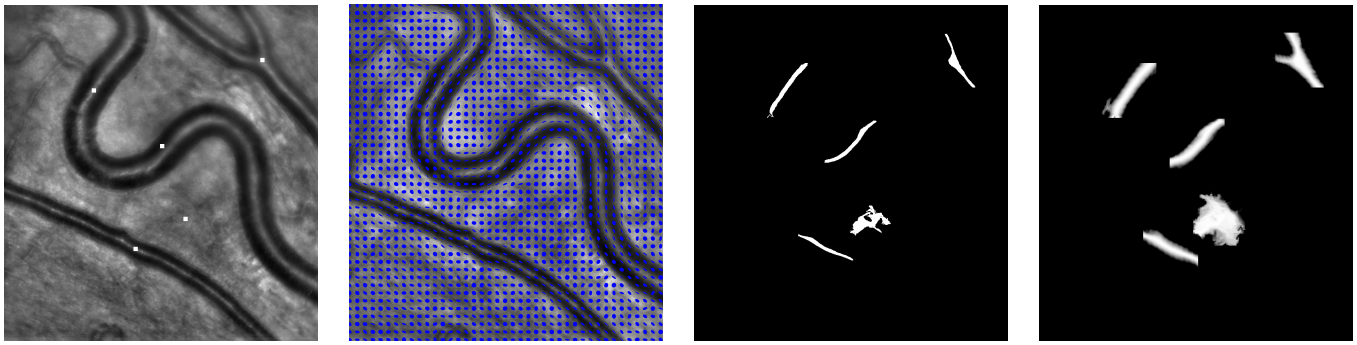
As shown in Section 3.1, Figures 1 and 2, anisotropy images are bright wherever an elongated shape is present in the corresponding original image. Therefore, bright pixels highlight macroscopic objects of interest such as vessels, as well as local anisotropic shapes caused by noise or less relevant objects. However, significant elongated shapes have an additional characteristic: the local orientations of the pixels composing them vary smoothly over large elongated neighborhoods. Hence, when connecting these pixels in the co-circularity graph and applying iterated erosions ε_W^k, as defined in Section 2.3 and with W the adjacency matrix of the graph, bright pixels belonging to relevant shapes tend to darken much later than those caused by noise (see Figure 8, top row).

Indeed, recall that ε_W^k roughly consists in assigning to each pixel the minimum value over an extended neighborhood in the co-circularity graph (at least in the binary case, and the non-binary case can be interpreted in terms of soft neighborhoods). For pixels belonging to relevant objects like vessels, these neighborhoods are precisely designed to match the underlying elongated shapes, even when k is large (see Figure 7). As these regions are consistently highly anisotropic, their minimum value is likely to be high. By contrast, for a pixel in the background (that is, outside any object of interest), large neighborhoods tend to be more isotropic and are more likely to include dark pixels (see one such example in Figure 7).

After the erosion ε_W^k of the anisotropic image, for k large enough, background pixels should result dark and brighter pixels should be found mainly in large anisotropic shapes. Therefore, subsequently applying the dilation δ_W^k allows us to recover most of the vessel shape. Indeed, this dilation can be seen as the diffusion of large values over the neighborhoods induced by the graph, and for pixels within the vessels the neighborhoods almost do not intersect the background. This can also be read in the light of the path interpretation given at the end of Section 2.

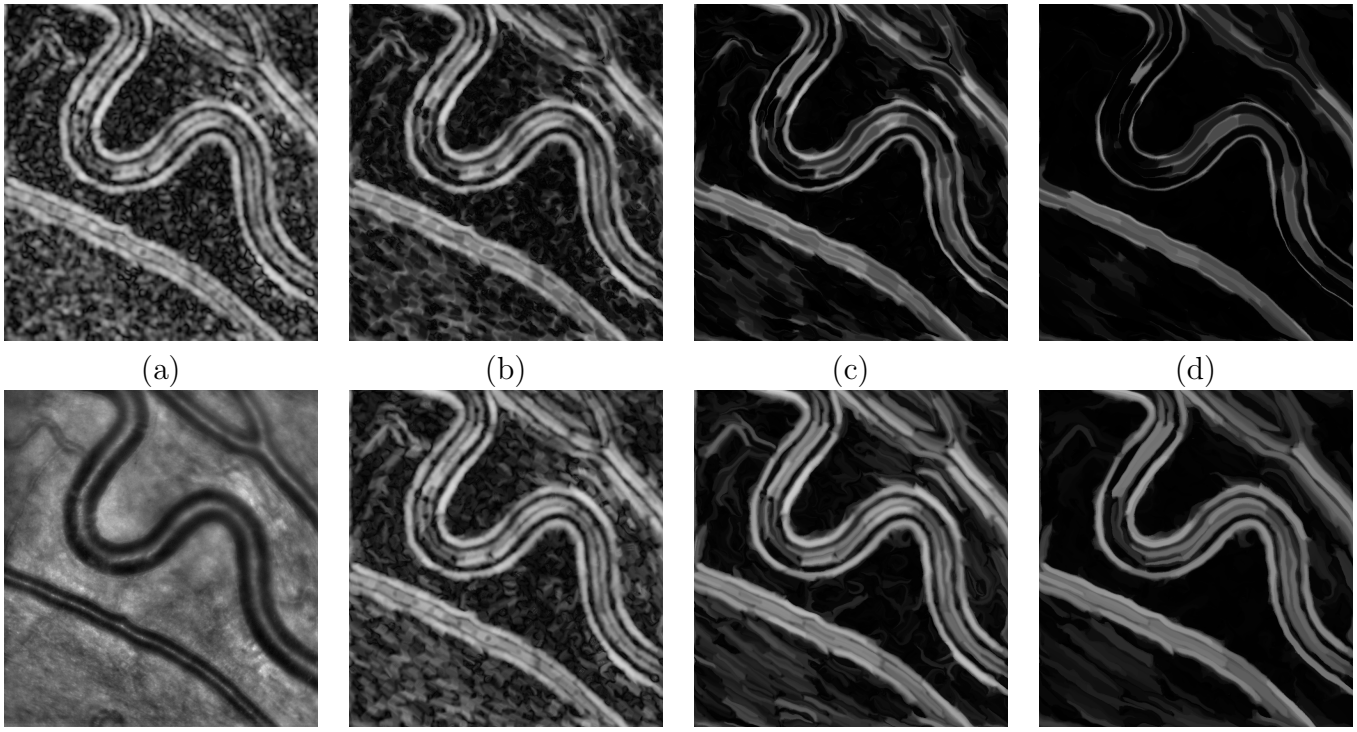
The results of openings $\gamma_{W^k} = \delta_W^k \varepsilon_W^k$ are shown in the bottom row of Figure 8 and in Figure 9. As we see in these examples the remaining bright pixels after filtering indicate the presence of a vessel. We shall note that large (quasi) flat regions follow geodesic paths, as predicted by the path interpretation of γ_{W^k}.

Applying a threshold to these outputs can already produce a fairly accurate mask of the vessels in the input image, and further processing may improve it. To conclude this section, in Figures 7 and 9 one may notice little difference between binary-weighted and non binary-weighted graphs. This will be discussed in more detail in Section 4.3.



(a) (b) (c) (d)

Figure 7: (a) Input image I with five pixels $i_1, i_2, \ldots, i_5$ marked as the center of the white squares. (b) A representation of the structure tensors T_σ for $\sigma = 5$. The corresponding anisotropy image $I_a^{(\sigma)}$ is shown in Figure 8 (a). (c) and (d): The neighborhoods $\mathcal{N}_{i_1}^k, \ldots, \mathcal{N}_{i_5}^k$ with $k = 10$, in the binary (c) and non-binary (d) cases. In both cases the co-circularity graph was obtained with the parameters $\sigma = 5$, $p = 7$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{20}$ in the binary case.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8: Erosions and openings of an anisotropy image. (a) Anisotropy image $I_a^{(\sigma)}$ computed from the input (e) for $\sigma = 5$. (b-d) From left to right: erosions $\varepsilon_W^k(I_a^{(\sigma)})$ for $k = 1, 5, 10$, with a binary-weighted co-circularity graph. (f-h) From left to right: openings $\gamma_{W^k}(I_a^{(\sigma)}) = \delta_W^k \varepsilon_W^k(I_a^{(\sigma)})$ for $k = 1, 5, 10$, with the same binary-weighted co-circularity graph. The graph was obtained with the parameters $\sigma = 5$, $p = 7$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{20}$.

4.2 Processing of Original Images

The morphological operators presented earlier, when applied to the original input image rather than the anisotropy one, show regularization properties: we observe a simplification of images with large flat areas, while most edges and elongated shapes are preserved. In Figures 10 and 11 we show the results of alternating sequential filters M_k defined by

$$M_k = \gamma_k \varphi_k \ldots \gamma_2 \varphi_2 \gamma_1 \varphi_1, \tag{30}$$

where we omitted W in the notations: $\gamma_i = \gamma_{W^i} = \delta_W^i \varepsilon_W^i$ and $\varphi_i = \varphi_{W^i} = \varepsilon_W^i \delta_W^i$ for $1 \leq i \leq k$.

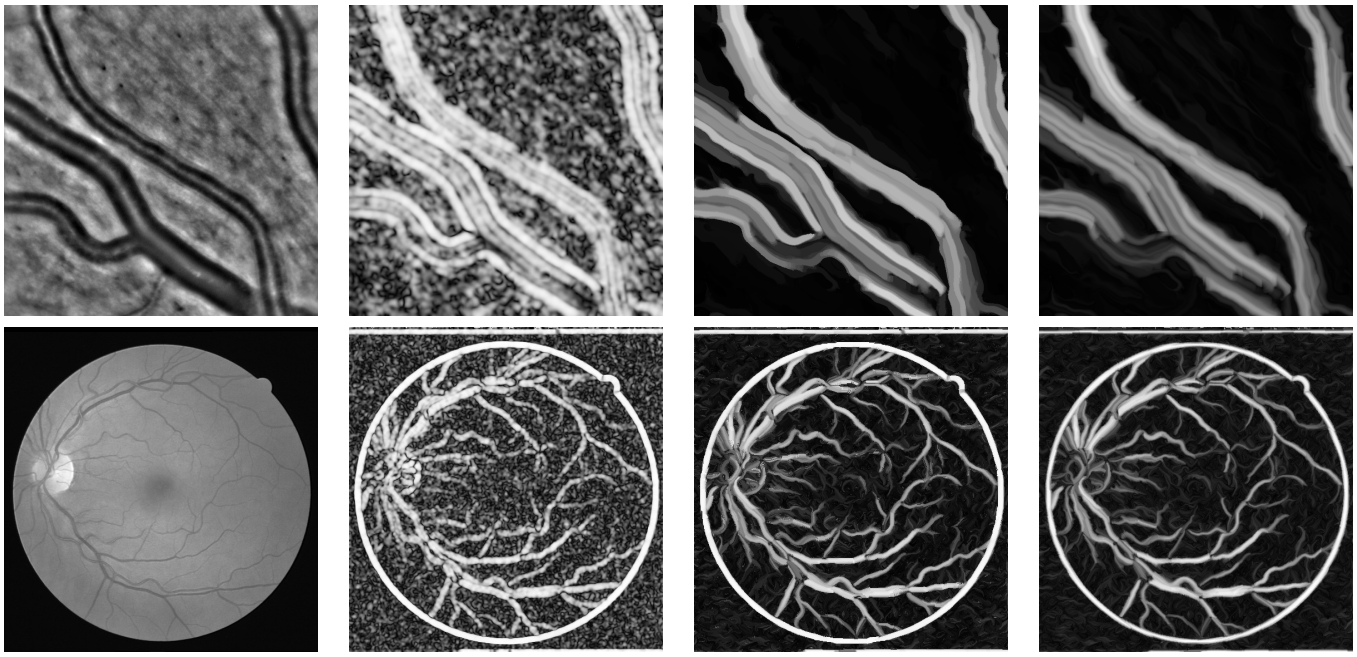


Figure 9: Openings of anisotropy images. Top row, from left to right: input image, anisotropy image $I_a^{(\sigma)}$ with $\sigma = 5$, openings $\gamma_{W^k}(I_a^{(\sigma)}) = \delta_W^k \varepsilon_W^k(I_a^{(\sigma)})$ for $k = 15$ in the binary and non-binary case. The co-circularity graphs were obtained with the parameters $\sigma = 5$, $p = 7$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{20}$ in the binary case. The bottom row follows the same pattern with a different input image and parameters: $k = 5$, $\sigma = 2$, $p = 3$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{12}$.

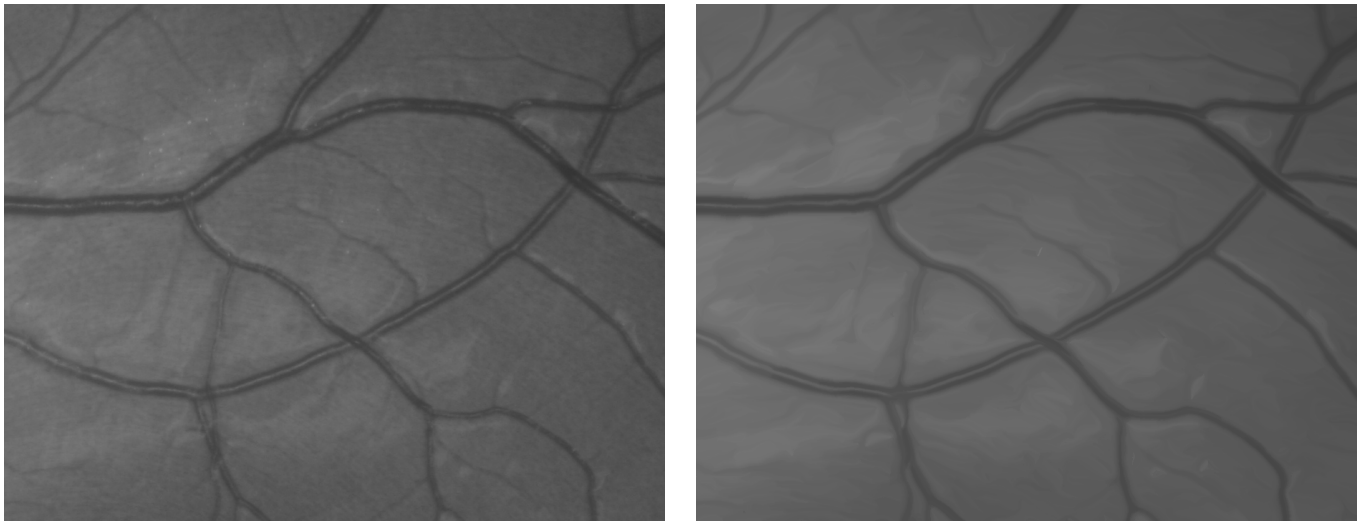


Figure 10: **Left:** input image, crop of a high resolution eye fundus. **Right:** result of alternating sequential filters M_k with $k = 3$ (see Equation (30)), based on a binary-weighted co-circularity graph whith parameters $\sigma = 5$, $p = 7$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{20}$. With the same parameters, the result with a non binary-weighted graph is very similar.

Whereas Figure 10 illustrates well the regularization property, Figure 11 shows that more textured regions without particular anisotropy can be significantly modified and some objects may disappear or have their aspect drastically changed. The overall result can be seen as the application of an impressionist painting-like style to the original image.

Another interesting application is the completion of interrupted edges in images, as shown in Figure 12. When two elongated shapes are in good continuation, like in the case of an edge which has been partially occluded, an anisotropic closing (for bright structures on a dark background) or an opening (for dark structures on a bright background) can fill the gap between the two adjacent

Figure 11: **Left:** input image. **Right:** result of alternating sequential filters M_k with $k = 5$ (see Equation (30)), based on a non binary-weighted co-circularity graph with parameters $\sigma = 2$, $p = 3$ and $\beta = \frac{\pi}{6}$.

extremities. To achieve this, the parameters σ and p must be adapted to the gap to bridge. Moreover if the distance separating the elements to join is large compared to their size, this completing effect may not be achievable by one closing, no matter its size (this might be clear from the interpretation of parameters provided in the next section). However, it is possible to compute a new graph from the output image, where extremities have been brought a little closer, and apply a new closing based on the new graph. Such iterations are obviously more computationally expensive.

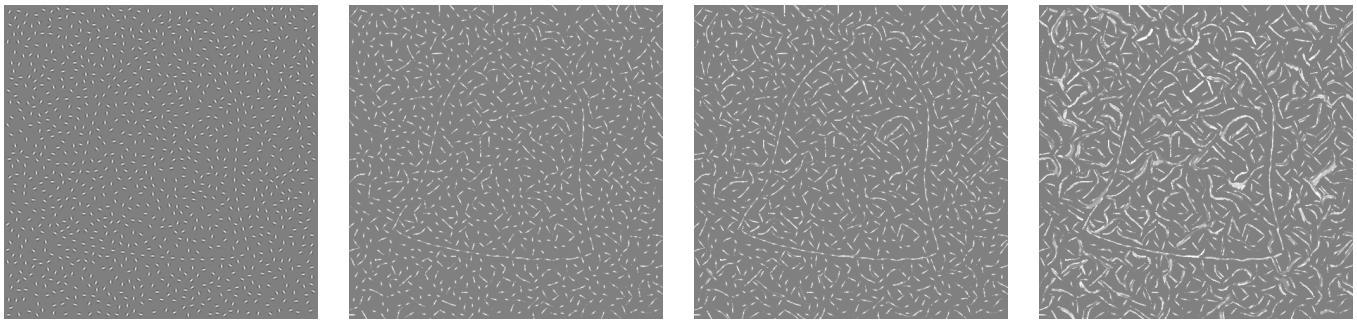


Figure 12: Input image and three closings $\varphi_{W^k} = \varepsilon_W^k \delta_W^k$ for $k = 1, 2, 3$ and W the adjacency matrix of the binary-weighted co-circularity graph obtained with parameters $\sigma = 4$, $p = 7$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{20}$.

4.3 Discussion

On the choice of σ. Roughly speaking, the scale parameter σ rules the size of the window over which we average zero-scale structure tensors T_0 to obtain T_σ. Therefore, the correlation between two structure tensors $T_\sigma(u_i)$ and $T_\sigma(u_j)$ increases with σ, while it decreases with the distance between u_i and u_j. As a consequence, the larger σ the more likely it is that neighboring tensors have similar orientations, and hence that every tensor has a co-circular one in its close neighborhood, that is, even with a small value of p. This is a desirable property as the effect of the morphological operators depends on the degree of connectivity of the vertices in the co-circularity graph. For example, in the case of edge completion, σ must be large enough to ensure that the tensors near the middle point between the extremities to join are roughly co-circular to the latter (provided, obviously, that p is also large enough for the extremities neighborhoods to cover the middle point).

On the other hand, σ should not exceed too much the dimensions of the objects of interest in the image (e.g. the width of vessels). If it does, the local anisotropy values and orientations carried by the structure tensors are not representative of underlying real objects. Typically, a too large value for σ would diffuse the anisotropy and orientation of a vessel far away from its boundaries.

Regarding the effect of σ on the operators activity, we shall notice that in general, for a fixed image and set of other parameters, the adjacency matrix does not increase with σ and neither does it decrease. This means that there is no inclusion relationship between the neighborhoods $\mathcal{N}_i^{\sigma_1}$ and $\mathcal{N}_i^{\sigma_2}$ of the same pixel (or vertex) i for $\sigma_1 < \sigma_2$. What is more, there is no ordering relationship between their surfaces - or integral in the non-binary case, that we define as

$$\mathcal{A}(i, W) = \sum_{j=0}^{n-1} w_{ij}. \tag{31}$$

In practice we observe that when p is large enough, a small σ leads to neighborhoods that are not smaller but can be less accurate with respect to underlying elongated shapes, as shown in Figure 13. This is due to the fact that with a small σ and a large p it is more likely that neighboring pixels be co-circular by chance and not because they belong to the same structure.

Figure 13: **Left:** Crop of the input image of Figure 7, with a pixel i marked as the center of the white square. **Center:** Neighborhood $\mathcal{N}_i^k$ in the non binary-weighted co-circularity graph with $k = 15$ and parameters $\sigma = 5$, $p = 7$ and $\beta = \frac{\pi}{6}$. **Right:** Neighborhood of the same pixel in the non binary-weighted graph with same parameters except $\sigma = 2$. As we can see the two neighborhoods are not ordered and the one for $\sigma = 2$ is less accurate with respect to the underlying vessel shape.

On the choice of p. The parameter p determines the square window around a pixel i, from which candidates to be neighbors in the co-circularity graphs will be chosen. As such, it is the strongest constraint on the connectivity of the graph, although we just saw that this also depends on σ. In binary-weighted graphs, the smaller p the more chances to find an isolated vertex or a small connected component that is disconnected from the rest of the graph, causing artifacts in the output of the morphological operators. This is even more so as σ is small. By contrast, when p is small, increasing σ can reduce this risk by increasing the correlation between neighbor orientations. However, if σ needs to be increased a lot, it might not be suited to the dimensions of the relevant objects in the image. Therefore in practice, in the occurrence of artifacts such as in Figures 14 and 15, increasing p is more efficient to make them disappear. Note that these artifacts are very unlikely with non binary-weighted graphs, as the disconnections observed in binary-weighted graphs are replaced by weak connections, for which the graph is usually connected.

The effect of p on the operators activity is more straightforward, as the adjacency matrix increases with p, everything else being fixed.

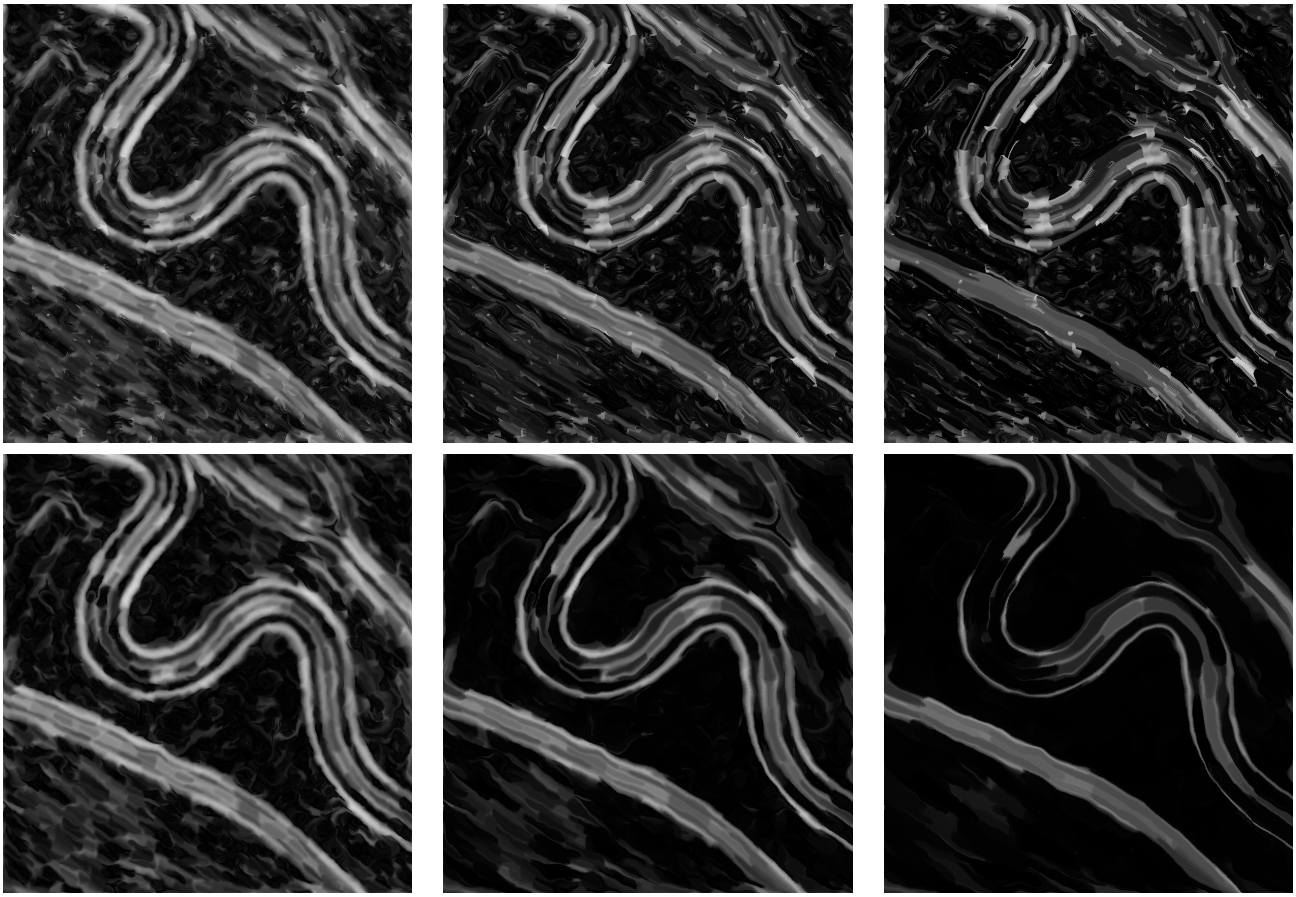


Figure 14: Illustration of the influence of parameter p on artifacts and operators activity. All images are binary-weighted erosions of the anisotropy image I_a^σ with $\sigma = 5$ (shown in Figure 8), but the parameter p is different between the top row ($p = 3$) and the bottom row ($p = 7$). **Top row:** $\varepsilon_W^k(I_a^\sigma)$ for $k = 5, 20, 70$. We see that the evolution is very slow, showing a weak activity of the erosions, and that artifacts appear because of many isolated connected components. **Bottom row:** $\varepsilon_W^k(I_a^\sigma)$ for $k = 2, 5, 10$. The evolution is quicker and no artifacts appear. Except for p and k, all other parameters are the same: $\sigma = 5$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{20}$.

On the choice of α and β. The angle β ruling the conic condition quantifies how much you allow elongated structures to "turn" on a maximal distance of p. Therefore, it determines the maximal curvature of shapes that operators will preserve and enhance. Increasing β increases the operators activity as the adjacency matrix increases with β. In practice, for all the applications presented here, we always used the same value $\beta = \frac{\pi}{6}$, which works fine. Other tested values showed that the method is quite robust with respect to the choice of β, as different values did not bring significant changes.

The angle α, which is used only to build binary-weighted graphs, is a precision tolerance quantifying how much we can stick to co-circularity. As such, it should be increased to process low resolution or noisy images, where there is more uncertainty on the estimation of local orientations. Like β and p, increasing α increases the operators activity. In our experiments, we almost always used $\alpha = \frac{\pi}{20}$ except for the low resolution eye fundus, for which we preferred $\alpha = \frac{\pi}{12}$ as $\alpha = \frac{\pi}{20}$ generated more artifacts.

On the choice between binary or non binary-weighted graphs. In Figures 7, 9 and 10 we emphasized the similarity of the results obtained with binary and non binary-weighted graphs. This is true in particular for $\alpha = \frac{\pi}{20}$ and the specific choice we made for a hidden parameter in the non-binary method: the value M in (27), which is the maximum possible pixel value ($M = 255$ for 8-bits images). A different constant could be chosen but this would have little effect provided we choose a

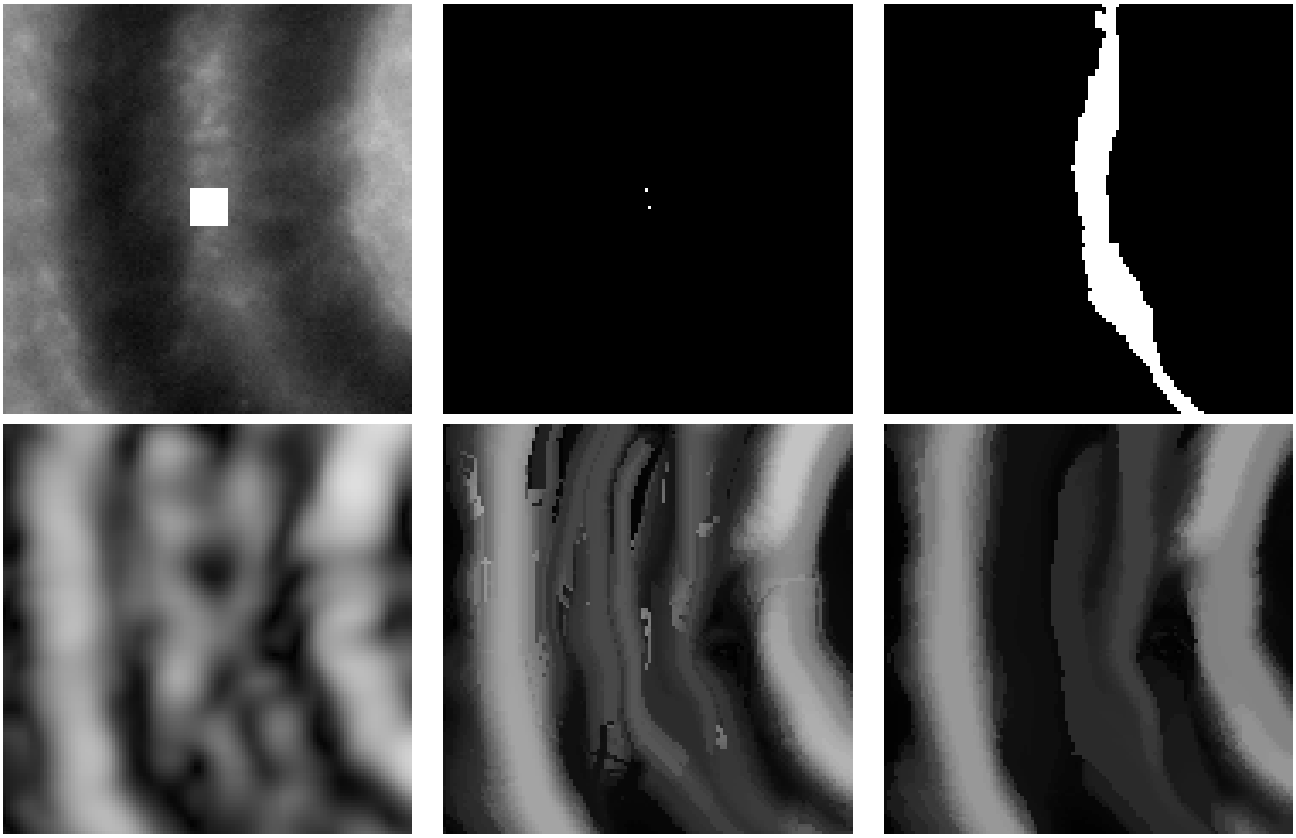


Figure 15: Illustration of artifacts due to a small value of p. **Top row, from left to right:** Crop of the input image of Figure 7, with a pixel i marked as the center of the white square; Neighborhood $\mathcal{N}_i^k$ in the binary-weighted co-circularity graph with $k = 10$ and parameters $\sigma = 5$, $p = 5$, $\beta = \frac{\pi}{6}$ and $\alpha = \frac{\pi}{20}$; Same, but with $p = 7$. When $p = 5$, $\mathcal{N}_i^k$ is restricted to only one pixel other than i. **Bottom row, from left to right:** crop of the anisotropy image for $\sigma = 5$, and openings γ_{W^k} with $k = 10$ using the two different co-circularity graphs ($p = 5$ and $p = 7$). The weak connectivity of pixel i when $p = 5$ causes an artifact which is not present when $p = 7$. Note that for $p = 5$ the parameter σ needs to be considerably increased before artifacts disappear.

constant with the same order of magnitude.

Indeed qualitatively, in both binary and non-binary cases, the edge weight w_{ij} is a function of an angle α_{ij}, the angle between the vectors $\mathbf{v}_i'$ and $\mathbf{v}_j$ - see Equations (25), (26) and (27). The smaller this angle, that is to say the closer we are to co-circularity between i and j, the closer to zero and the more similar the weights in the binary and non-binary case. Since the weights that contribute the most to the results of the morphological operators are precisely those close to zero, it is not surprising that binary and non-binary co-circular graphs yield similar results in many cases.

The main difference appears precisely where configurations deviate from co-circularity and the larger weights are not so close to zero. As we saw, this creates isolated connected components and artifacts in the binary case, whereas it does not happen in the non binary case where disconnections are replaced by soft weak connections. Therefore, opting for the non binary-weighted graph is a good way to avoid artifacts without the need to adjust the other parameters, e.g. increasing p.

The disadvantage of the non-binary formulation is that it is more expensive computationally than the binary one. Although the theoretical complexity is the same, as we will see next, the matrix W is less sparse. As a consequence, more values are assigned when computing the matrix, and more comparisons are made in the application of the morphological operator. This is why we often work with the binary formulation in practice, especially with large images, unless we do not quickly find satisfactory parameters, which avoid artifacts and fit the physical reality of underlying objects.

On the complexity of Algorithms 1, 2 and 3. Given an image with n pixels, the computation of the adjacency matrix representing the co-circularity graph (Algorithm 1) consists in testing, for every pixel, its adjacency with the pixels composing its $(2p+1)\times(2p+1)$ square neighborhood. As this test requires a number of operations upper-bounded by a constant independent from n and p, the complexity of Algorithm 1 is $\mathcal{O}(np^2)$.

The computation of the dilation $\delta_W(I)$ (Algorithm 2) is linear in the number of finite coefficients in the sparse matrix W, which is at most $n(2p+1)^2$. Hence the complexity of the dilation is at most $\mathcal{O}(np^2)$. The complexity of the erosion (Algorithm 3) is the same.

Therefore, the complexity of computing the adjacency graph and then applying morphological operators based on iterated erosions and dilations is $\mathcal{O}(np^2)$. As we pointed out earlier, in practice the non-binary case is more expensive, as it is closer to the worst case, than the binary one.

On the limitations of the approach. The main drawback of the approach presented in this paper is that it is mono-scale, and therefore not well suited for images where elongated shapes appear at very different scales. Multi-scale adaptations can be thought of, as for example by replacing the one-scale adjacency matrix W_σ by the supremum of adjacency matrices over a range of scales Σ, $W = \bigvee_{\sigma\in\Sigma} W_\sigma$. When processing the anisotropy image, the latter could also be replaced by an integration of anisotropy images of different scales, for example by supremum.

Another limitation is that the method relies on an estimation of local orientations, which becomes inaccurate when the image resolution decreases. Therefore, our approach seems more suited to high resolution images.

Finally, so far this framework only allows processing grayscale images. While this might be sufficient for shape analysis, it would be highly desirable to extend the framework to color images for the regularization and style application aspects.

5 Conclusion

We presented in this paper in full detail an algorithm for morphological anisotropic processing of grayscale images, which had only been briefly introduced in previous work. In the present extension, several examples of applications, a discussion on the parameters and the release of the source code are meant to provide a thorough understanding of the method and help users make the best of it in practical cases as well as in methodological studies. Furthermore, besides recalling the tropical formulation of adaptive non-flat morphological operators introduced earlier, here we added a new theoretical contribution through the interpretation of openings in terms of paths in a graph. Finally, we pointed out several tracks to extend and improve our approach.

A Pseudo-code

B Proof of Proposition 3

The proof is the direct application of the definitions of Equation (8)

$$\begin{aligned}
\gamma_W(\mathbf{x})_i \geq t \iff \delta_W\varepsilon_W(\mathbf{x})_i \geq t \quad &\iff \quad \textstyle\bigvee_{j\in\mathcal{N}_i} \varepsilon_W(x)_j + w_{ij} \geq t \\
&\iff \quad \exists j \in \mathcal{N}_i, \varepsilon_W(x)_j \geq t - w_{ij} \\
&\iff \quad \exists j \in \mathcal{N}_i, \textstyle\bigwedge_{l\in\check{\mathcal{N}}_j} x_l - w_{lj} \geq t - w_{ij} \\
&\iff \quad \exists j \in \mathcal{N}_i, \forall l \in \check{\mathcal{N}}_j\ x_l \geq t - w_{ij} + w_{lj}.
\end{aligned}$$

Pseudo-code	Source code	Souce file
Algorithm 1, ImageToGraph()	`image_to_cocircularity_graph()`	`cocircularity_graphs.cpp`
ComputeAdjacencyMatrix()	`compute_adjacency_matrix()`	`cocircularity_graphs.cpp`
TestAdjacency()	`test_adjacency()`	`cocircularity_graphs.cpp`
Algorithm 2, Dilation()	`dilate_max_plus_symmetric()`	`max_plus_operators.cpp`
Algorithm 3, Erosion()	`erode_max_plus_symmetric()`	`max_plus_operators.cpp`
Gradient()	`lin_grad_gray()`	`structure_tensors.cpp`
StructureTensors0()	`structure_tensors0()`	`structure_tensors.cpp`
SmoothStructureTensors()	`smooth_structure_tensors()`	`structure_tensors.cpp`
DiagonalizeStructureTensors()	`diagonalize_structure_tensors()`	`structure_tensors.cpp`
GaussianBlur()	`gaussian_filter()`	`structure_tensors.cpp`

Table 1: Correspondance between pseudo-code and source code.

Algorithm 1: ImageToGraph$(I, \sigma, p, cos_\alpha, cos_\beta, bin)$

Input: A two dimensional array I representing a grayscale image of width $\mathcal{W}$ and height $\mathcal{H}$; a positive real number σ representing the standard deviation of the isotropic Gaussian kernel; a positive integer p defining a $(2p+1) \times (2p+1)$ square neighborhood; the cosine of an angle $\alpha \in [0, \frac{\pi}{2})$ representing the angular precision to relax the co-circularity constraint; the cosine of an angle $\beta \in [0, \frac{\pi}{2})$ to define the conic condition; a boolean bin to indicate if the graph should be binary-weighted ($bin = True$) or non binary-weighted ($bin = False$).

Output: A two dimensional array $I_a^{(\sigma)}$ representing the anisotropy image ; a symmetric sparse matrix W representing the adjacency matrix of the graph.

$dI_x, dI_y \leftarrow$ `Gradient`(I); /* *Compute gradient, see Appendix A.* */

$a_0, b_0, c_0 \leftarrow$ `StructureTensors0`(dI_x, dI_y); /* *Compute T_0; see Equation (18) and Appendix A.* */

$a_\sigma, b_\sigma, c_\sigma \leftarrow$ `SmoothStructureTensors`(a_0, b_0, c_0, σ); /* *Compute T_σ; see Equation (19) and Appendix A.* */

$I_a^{(\sigma)}, \mathbf{v} \leftarrow$ `DiagonalizeStructureTensors`$(a_\sigma, b_\sigma, c_\sigma)$; /* *Anisotropy and local orientations; see Appendix A.* */

$W \leftarrow$ `ComputeAdjacencyMatrix`$(\mathbf{v}, p, cos_\beta, bin, \cos_\alpha)$; /* *Adjacency matrix.* */

return $I_a^{(\sigma)}, W$;

Function ComputeAdjacencyMatrix($\mathbf{v}, p, cos_\beta, bin, cos_\alpha$)

Input: A list $\mathbf{v} = (\mathbf{v}_i)_{0 \leq i \leq n-1}$ of $n = \mathcal{W} \cdot \mathcal{H}$ unitary vectors representing the main orientations at each pixel of a $\mathcal{W} \times \mathcal{H}$ image; a positive integer p defining a $(2p+1) \times (2p+1)$ square neighborhood; the cosine of an angle $\beta \in [0, \frac{\pi}{2})$ to define the conic condition; a boolean bin to indicate if the graph should be binary-weighted ($bin = True$) or non binary-weighted ($bin = False$); the cosine of an angle $\alpha \in [0, \frac{\pi}{2})$ representing the angular precision to relax the co-circularity constrain (binary case, used only if $bin = True$).

Output: A sparse symmetric $n \times n$ matrix W.

```
W ← −∞;                                          /* Initialize all coefficients to −∞. */
for c ← 0 to H − 1 do
    for r ← 0 to W − 1 do
        i ← r × W + c;
        foreach (r′, c′) in R(u_i, p)            /* R(u_i, p) as defined in Equation (29). */ do
            j ← r′ × W + c′;
            W_ij ← TestAdjacency(u_i, v_i, u_j, v_j, cos_β, bin, cos_α);
            W_ji ← W_ij;
        end
    end
end
return (W);
```

Algorithm 2: Dilation(I, W)

Input: A $\mathcal{W} \times \mathcal{H}$ grayscale image I; a sparse $n \times n$ matrix W, where $n = \mathcal{W} \cdot \mathcal{H}$.

Output: The dilated image I_{dil} and a boolean variable *change* indicating if $I \neq I_{dil}$ ($change = True$) or $I = I_{dil}$ ($change = False$)

```
Define T := {(i, j, w), w_ij = w > −∞};
I_dil ← copy(I);
foreach (i, j, w) ∈ T do
    (r_i, c_i) ← (⌊i/W⌋, i − ⌊i/W⌋);
    (r_j, c_j) ← (⌊j/W⌋, j − ⌊j/W⌋);
    if I_dil[r_i, c_i] < I[r_j, c_j] + w then
        I_dil[r_i, c_i] ← I[r_j, c_j] + w;
        change = True;
    end
end
return I_dil, change
```

Algorithm 3: Erosion(I, W)

Input: A $\mathcal{W} \times \mathcal{H}$ grayscale image I; a sparse $n \times n$ matrix W, where $n = \mathcal{W} \cdot \mathcal{H}$.
Output: The eroded image I_{ero} and a boolean variable *change* indicating if $I \neq I_{ero}$ ($change = True$) or $I = I_{ero}$ ($change = False$)

```
Define T := {(i, j, w), w_ij = w > −∞};
I_ero ← copy(I);
foreach (i, j, w) ∈ T do
    (r_i, c_i) ← (⌊i/W⌋, i − ⌊i/W⌋);
    (r_j, c_j) ← (⌊j/W⌋, j − ⌊j/W⌋);
    if I_ero[r_i, c_i] > I[r_j, c_j] − w then
        I_ero[r_i, c_i] ← I[r_j, c_j] − w;
        change = True;
    end
end
return I_ero, change
```

Function TestAdjacency$(u_1, \mathbf{v}_1, u_2, \mathbf{v}_2, cos_\beta, bin, cos_\alpha)$

Input: Two pixel positions $u_1 = (r_1, c_1)$ and $u_2 = (r_2, c_2)$ and two unitary vectors $\mathbf{v}_1 = [\cos(\theta_1), \sin(\theta_1)]$ and $\mathbf{v}_2 = [\cos(\theta_2), \sin(\theta_2)]$ representing the main orientations at pixels u_1 and u_2 respectively; the cosine of an angle $\beta \in [0, \frac{\pi}{2})$ to define the conic condition; a boolean *bin* to indicate if the graph should be binary-weighted ($bin = True$) or non binary-weighted ($bin = False$); the cosine of an angle $\alpha \in [0, \frac{\pi}{2})$ representing the angular precision to relax the co-circularity constrain (binary case, used only if $bin = True$).
Output: An adjacency weight w associated to $\Big((u_1, \theta_1), (u_2, \theta_2)\Big)$, as defined by Equation (25)-(27).

```
w ← −∞;
d ← (u_2 − u_1) / ||u_2 − u_1||_2;                    /* See Figure 4. */
if |d · v_1| ≥ cos_β                                  /* Conic condition (Equation (24)). */ then
    v'_1 ← 2(v_1 · d)d − v_1;                         /* See Figure 4. */
    if bin then
        resp ← 1_[cos_α,1](|v'_1 · v_2|);             /* Binary case. */
    else
        resp ← |v'_1 · v_2|;                          /* Non-binary case. */
    end
    w ← M · log(resp);                                /* M being the maximal possible image intensity. */
end
return w;
```

Function Gradient(I)

Input: A two dimensional array I representing a grayscale image of width $\mathcal{W}$ and height $\mathcal{H}$.
Output: Two arrays dI_x and dI_y representing the two components of the gradient ∇I at each pixel.

```
/* Compute dI_x                                                    */
for i ← 0 to H − 1 do
    dI_x[i, 0] ← I[i, 1] − I[i, 0] ;                   /* Case j = 0.  */
    dI_x[i, W − 1] ← I[i, W − 1] − I[i, W − 2] ;       /* Case j = W − 1. */
    for j ← 1 to W − 2 do
        dI_x[i, j] ← 0.5 × (I[i, j + 1] − I[i, j − 1]);
    end
end
/* Compute dI_y                                                    */
for j ← 0 to W − 1 do
    dI_y[0, j] ← I[1, j] − I[0, j] ;                   /* Case i = 0.  */
    dI_y[H − 1, j] ← I[H − 1, j] − I[H − 2, j] ;       /* Case i = H − 1. */
    for i ← 1 to H − 2 do
        dI_y[i, j] ← 0.5 × (I[i + 1, j] − I[i − 1, j]);
    end
end
return dI_x, dI_y;
```

Function StructureTensors0(dI_x, dI_y)

Input: Two arrays dI_x and dI_y representing the two components of the gradient ∇I of a grayscale image I of width $\mathcal{W}$ and height $\mathcal{H}$.
Output: Three arrays a_0, b_0 and c_0 representing the components that fully determine the 2×2 symmetric tensor $T_0 = \begin{pmatrix} a_0 & c_0 \\ c_0 & b_0 \end{pmatrix}$, at each pixel, as defined by Equation (18).

```
for i ← 0 to H − 1 do
    for j ← 0 to W − 1 do
        a_0[i, j] ← dI_x[i, j]^2;
        b_0[i, j] ← dI_y[i, j]^2;
        c_0[i, j] ← dI_x[i, j] × dI_y[i, j];
    end
end
return a_0, b_0, c_0;
```

Function SmoothStructureTensors(a_0, b_0, c_0, σ)

Input: A positive real number σ representing the standard deviation of the isotropic Gaussian kernel ; Three arrays a_0, b_0 and c_0 representing the components of a 2×2 symmetric tensor $T_0 = \begin{pmatrix} a_0 & c_0 \\ c_0 & b_0 \end{pmatrix}$, at each pixel.

Output: Three arrays a_σ, b_σ and c_σ representing the components that fully determine the 2×2 symmetric tensor $T_\sigma = \begin{pmatrix} a_\sigma & c_\sigma \\ c_\sigma & b_\sigma \end{pmatrix}$ as defined by Equation (19).

$a_\sigma \leftarrow$ `GaussianBlur`(a_0, σ);
$b_\sigma \leftarrow$ `GaussianBlur`(b_0, σ);
$c_\sigma \leftarrow$ `GaussianBlur`(c_0, σ);
return $a_\sigma, b_\sigma, c_\sigma$;

Function DiagonalizeStructureTensors(a, b, c)

Input: Three arrays a, b and c representing the components of a $\mathcal{W} \times \mathcal{H}$ array T containing 2×2 symmetric tensors $T = \begin{pmatrix} a & c \\ c & b \end{pmatrix}$.

Output: An array I_a representing the anisotropy image associated to the input structure tensors, as defined by Equation (22), of width $\mathcal{W}$ and height $\mathcal{H}$; a set $\mathbf{v}$ of normalized (or null) vectors corresponding to the second eigenvectors of the input structure tensors.

```
for i ← 0 to H − 1 do
    for j ← 0 to W − 1 do
        Δ ← (a[i,j] − b[i,j])^2 + 4c[i,j]^2;
        Trace ← a[i,j] + b[i,j];
        λ1 ← 0.5 × (a[i,j] + b[i,j] + √Δ);
        λ2 ← 0.5 × (a[i,j] + b[i,j] − √Δ);
        if Trace = 0 then
            I_a[i,j] ← 0;
        else
            I_a[i,j] ← 1 − 2 λ2/Trace;
        end
        x1 ← 2c[i,j];
        x2 ← b[i,j] − a[i,j] − √Δ;
        norm_ev ← √(x1^2 + x2^2);
        if norm_ev > 0 then
            v_x[i,j] ← x1/norm_ev;
            v_y[i,j] ← x2/norm_ev;
        else
            v_x[i,j] ← 0;
            v_y[i,j] ← 0;
        end
    end
end
return I_a, v;
```

Function GaussianBlur(I, σ)

Input: A two-dimensional array I of width $\mathcal{W}$ and height $\mathcal{H}$; a positive real number σ representing the standard deviation of an isotropic Gaussian kernel

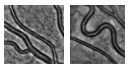
Output: A two dimensional array I_{out} representing the smoothed version of I, obtained by convolution with the Gaussian kernel

```
ρ ← ⌈2σ⌉;                    /* Radius that defines the size of the kernel: (2ρ+1) × (2ρ+1) */
/* Convolution along the width (rows, or x dimension) */
for i ← 0 to H − 1 do
    for j ← 0 to W − 1 do
        sumV ← 0;            /* To store the dot product between kernel and neighborhood. */
        sumK ← 0;            /* To store the sum of the kernel weights. */
        for k ← −ρ to ρ do
            if 0 ≤ j + k ≤ W − 1 then
                valK ← exp(−k²/(2σ²));
                sumV ← sumV + I[i, j + k] × valK;
                sumK ← sumK + valK;
            end
        end
        I_X[i, j] ← sumV/sumK;
    end
end
/* Convolution along the height (columns, or y dimension) */
for i ← 0 to H − 1 do
    for j ← 0 to W − 1 do
        sumV ← 0;            /* To store the dot product between kernel and neighborhood. */
        sumK ← 0;            /* To store the sum of the kernel weights. */
        for k ← −ρ to ρ do
            if 0 ≤ i + k ≤ H − 1 then
                valK ← exp(−k²/(2σ²));
                sumV ← sumV + I_X[i + k, j] × valK;
                sumK ← sumK + valK;
            end
        end
        I_out[i, j] ← sumV/sumK;
    end
end
return I_out;
```

Acknowledgment

This work was partly supported by a grant Inter-Carnot M.I.N.E.S. - TSN CARNOT 2016 - 60656, and was partially done while S. Blusseau and I. Bloch were with LTCI, Télécom Paris, Paris, France.

Image Credits

 Adaptive optic images of retina vessels, courtesy of Professor Michel Paques, Clinical Investigation Center 1423, Quinze-Vingts Hospital, Paris, France.

First test image of the DRIVE database [28][3], converted to a png grayscale image. ©IEEE.

Crop from the fourth image of the healthy set of the High-Resolution Fundus (HRF) Image Database [4][4], converted to a png grayscale image. Creative Commons 4.0 Attribution License https://creativecommons.org/licenses/by/4.0/.

Stimulus image from the study [24]. Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License http://www.creativecommons.org/licenses/by-nc-nd/3.0/.

Val D'Orcia In Autumn[5] by Fabrizio Lunardi, converted to a png grayscale image. Creative Commons CC0 1.0 Universal Public Domain Dedication https://creativecommons.org/publicdomain/zero/1.0/deed.en.

All other images produced by the authors.

References

[1] M. Akian, R. Bapat, and S. Gaubert, *Max-plus algebra*, Handbook of linear algebra (Discrete Mathematics and its Applications), 39 (2006), pp. 10–14.

[2] I. Bloch, H.J.A.M. Heijmans, and C. Ronse, *Mathematical morphology*, in Handbook of Spatial Logics, M. Aiello, I. Pratt-Hartmann, and J. Van Benthem, eds., Springer Netherlands, Dordrecht, 2007, pp. 857–944. https://doi.org/10.1007/978-1-4020-5587-4_14.

[3] S. Blusseau, S. Velasco-Forero, J. Angulo, and I. Bloch, *Tropical and morphological operators for signal processing on graphs*, in IEEE International Conference on Image Processing, 2018, pp. 1198–1202. https://doi.org/10.1109/ICIP.2018.8451395.

[4] A. Budai, R. Bock, A. Maier, J. Hornegger, and G. Michelson, *Robust vessel segmentation in fundus images*, International Journal of Biomedical Imaging, (2013), pp. 1–11. http://dx.doi.org/10.1155/2013/154860.

[5] B. A. Carré, *An algebra for network routing problems*, IMA Journal of Applied Mathematics, 7 (1971), pp. 273–294.

[6] M. Charif-Chefchaouni and D. Schonfeld, *Spatially-Variant Mathematical Morphology*, Springer US, Boston, MA, 1996, pp. 49–56. https://doi.org/10.1007/978-1-4613-0469-2_7.

[3] https://drive.grand-challenge.org/
[4] https://www5.cs.fau.de/research/data/fundus-images/
[5] https://commons.wikimedia.org/wiki/File:Val_D_Orcia_In_Autumn_(179351679).jpeg

[7] F. Cheng and A.N. Venetsanopoulos, *Adaptive morphological operators, fast algorithms and their applications*, Pattern Recognition, 33 (2000), pp. 917–933. https://doi.org/10.1016/S0031-3203(99)00155-7.

[8] R. A. Cuninghame-Green, *Minimax algebra*, vol. 166 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag Berlin Heidelberg, 1979.

[9] J. Debayle and J.C. Pinoli, *Spatially adaptive morphological image filtering using intrinsic structuring elements*, Image Analysis & Stereology, 24 (2005), pp. 145–158. https://doi.org/10.5566/ias.v24.p145-158.

[10] ——, *General adaptive neighborhood image processing: Part I: Introduction and Theoretical Aspects*, Journal of Mathematical Imaging and Vision, 25 (2006), pp. 245–266. https://doi.org/10.1007/s10851-006-7451-8.

[11] W. Förstner, *A feature based correspondence algorithm for image matching*, in Symposium of the International Society for Photogrammetry and Remote Sensing, Commission III, 1986, pp. 150–166.

[12] C. Harris and M. Stephens, *A combined corner and edge detector*, in Alvey vision conference, vol. 15, 1988, pp. 10–5244.

[13] H.J.A.M. Heijmans, *Composing morphological filters*, IEEE Transactions on Image Processing, 6 (1997), pp. 713–723. https://doi.org/10.1109/83.568928.

[14] H.J.A.M. Heijmans, M. Buckley, and H. Talbot, *Path openings and closings*, Journal of Mathematical Imaging and Vision, 22 (2005), pp. 107–119. https://doi.org/10.1007/s10851-005-4885-3.

[15] H.J.A.M. Heijmans and C. Ronse, *The algebraic basis of mathematical morphology I. Dilations and erosions*, Computer Vision, Graphics, and Image Processing, 50 (1990), pp. 245 – 295. http://dx.doi.org/10.1016/0734-189X(90)90148-O.

[16] G. Kanizsa, *Organization in vision: Essays on Gestalt perception*, Praeger Publishers, 1979.

[17] M. Kass and A. Witkin, *Analyzing oriented patterns*, Computer Vision, Graphics, and Image Processing, 37 (1987), pp. 362–385. https://doi.org/10.1016/0734-189X(87)90043-0.

[18] R. Lerallut, E. Decencière, and F. Meyer, *Image filtering using morphological amoebas*, Image and Vision Computing, 25 (2007), pp. 395–404. https://doi.org/10.1016/j.imavis.2006.04.018.

[19] G. L. Litvinov, *Maslov dequantization, idempotent and tropical mathematics: A brief introduction*, Journal of Mathematical Sciences, 140 (2007), pp. 426–444. https://doi.org/10.1007/s10958-007-0450-5.

[20] O. Merveille, B. Naegel, H. Talbot, L. Najman, and N. Passat, *2D Filtering of Curvilinear Structures by Ranking the Orientation Responses of Path Operators (RORPO)*, Image Processing On Line, 7 (2017), pp. 246–261. https://doi.org/10.5201/ipol.2017.207.

[21] O. Merveille, H. Talbot, L. Najman, and N. Passat, *Curvilinear structure analysis by ranking the orientation responses of path operators*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 40 (2018), pp. 304–317. https://doi.org/10.1109/TPAMI.2017.2672972.

[22] C. Ronse, *Why mathematical morphology needs complete lattices*, Signal Processing, 21 (1990), pp. 129 – 154. https://doi.org/10.1016/0165-1684(90)90046-2.

[23] P. Salembier, *Study on nonlocal morphological operators*, in IEEE International Conference on Image Processing, 2009, pp. 2269–2272. https://doi.org/10.1109/ICIP.2009.5414374.

[24] M. Sassi, K. Vancleef, B. Machilsen, S. Panis, and J. Wagemans, *Identification of Everyday Objects on the Basis of Gaborized Outline Versions*, i-Perception, 1 (2010), pp. 121–142. https://doi.org/10.1068/i0384.

[25] J. Serra, *Image analysis and mathematical morphology*, Academic Press, Inc., 1982.

[26] ——, *Image Analysis and Mathematical Morphology: Vol.: 2: Theoretical Advances*, Academic Press, London, 1988.

[27] J. Serra and L. Vincent, *An overview of morphological filtering*, Circuits, Systems and Signal Processing, 11 (1992), pp. 47–108. https://doi.org/10.1007/BF01189221.

[28] 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 (2004), pp. 501–509. https://doi.org/10.1109/TMI.2004.825627.

[29] O. Tankyevych, H. Talbot, and P. Dokladal, *Curvilinear morpho-Hessian filter*, in IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008, pp. 1011–1014. https://doi.org/10.1109/ISBI.2008.4541170.

[30] S. Velasco-Forero and J. Angulo, *On nonlocal mathematical morphology*, in International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, Springer, 2013, pp. 219–230. https://doi.org/10.1007/978-3-642-38294-9_19.

[31] ——, *Nonlinear operators on graphs via stacks*, in Geometric Science of Information, Proceedings, F. Nielsen and F. Barbaresco, eds., Springer International Publishing, 2015, pp. 654–663. https://doi.org/10.1007/978-3-319-25040-3_70.

[32] R. Verdú-Monedero, J. Angulo, and J. Serra, *Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields*, IEEE Transactions on Image Processing, 20 (2011), pp. 200–212. https://doi.org/10.1109/TIP.2010.2056377.

[33] S. Yang and J.X. Li, *Spatial-variant morphological filters with nonlocal-patch-distance-based amoeba kernel for image denoising*, Image Analysis & Stereology, 34 (2015), pp. 63–72. https://doi.org/10.5566/ias.1098.