Universitat
de les Illes Balears

DOCTORAL THESIS

2018

Curvilinear Object Detection with Fuzzy Mathematical Morphology for Grayscale and Color Medical Imagery

Pedro BIBILONI SERRANO

DOCTORAL THESIS

2018

Doctoral Programme in Information and Communications Technology

CURVILINEAR OBJECT DETECTION WITH FUZZY
MATHEMATICAL MORPHOLOGY FOR GRAYSCALE AND
COLOR MEDICAL IMAGERY

Pedro BIBILONI SERRANO

Supervisors: Dr. Manuel GONZÁLEZ HIDALGO, and
Dr. Sebastià MASSANET MASSANET

Tutor: Dr. Sebastià MASSANET MASSANET

Doctor by the Universitat de les Illes Balears

Pedro BIBILONI **S**ERRANO
Curvilinear Object Detection with Fuzzy Mathematical Morphology for Grayscale and Color Medical Imagery
Autumn 2018
Supervisors: Dr. Manuel G ONZÁLEZ HIDALGO and Dr. Sebastià MASSANET MASSANET

Universitat de les Illes Balears
Departament de Ciències Matemàtiques i Informàtica
SCOPIA Research group

UIB

Universitat
de les Illes Balears

I, Pedro BIBILONI SERRANO, declare that this thesis titled, "*Curvilinear Object Detection with Fuzzy Mathematical Morphology for Grayscale and Color Medical Imagery*" and the work presented in it are my own. I confirm that:

– This work was done wholly or mainly while in candidature for a Ph.D. degree at this University.

– Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.

– Where I have consulted the published work of others, this is always clearly attributed.

– Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

– I have acknowledged all main sources of help.

– Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

For all intents and purposes, I hereby sign this document.

Signed:
Palma, Autumn 2018.

Universitat
de les Illes Balears

Dr. Manuel GONZÁLEZ HIDALGO of the Universitat de les Illes Balears, and Dr. Sebastià MASSANET MASSANET of the Universitat de les Illes Balears, declare that the thesis titled "*Curvilinear Object Detection with Fuzzy Mathematical Morphology for Grayscale and Color Medical Imagery*", presented by Pedro BIBILONI SERRANO to obtain a doctoral degree, has been completed under our supervision and meets the requirements to opt for an International Doctorate.
For all intents and purposes, we hereby sign this document.

Signed: Dr. Manuel GONZÁLEZ HIDALGO
Palma, Autumn 2018.

Signed: Dr. Sebastià MASSANET MASSANET
Palma, Autumn 2018.

Abstract

Fuzzy mathematical morphology is a set of tools to process grayscale images. It is based on two operators, the dilation and the erosion, that respectively enlarge and shrink objects. We extend these operators to deal with multivariate images by defining the soft color dilation and the soft color erosion. They are designed for generic multivariate color spaces, but also to process natural images consistently with regard to the notions of *enlarging* and *shrinking* objects. Besides being able to preserve colors, other theoretical properties are transferred from the fuzzy mathematical morphology. The soft color dilation and erosion can also be combined, in the same way as the fuzzy erosion and dilation, to provide operators with a complex behaviour. Several of such combinations have been designed for a variety of tasks, and can now be transferred to color images: noise filtering, contrast enhancing, object segmentation and shape recognition, among others. In this thesis, we also propose a definition of curvilinear objects to unify the literature: several image processing problems consider the task of segmenting tubular-shaped objects clearly different to their surrounding background. In particular, we study such problems to extract their common denominator. This state of the art is synthesized by categorizing both the approaches to segment curvilinear objects and the features they consider of interest. Besides, we design algorithms based on morphological operators to segment curvilinear objects. We use fuzzy mathematical morphology to segment vessels in eye-fundus photographs and soft color morphology to detect hair in dermoscopic images. Both morphologies consider different implementations of erosion and dilation. However, the dilation and erosion of each morphology can be combined similarly. Both methods achieve high performance compared to other published works. This has several implications: first, it indicates that the soft color morphology is a comprehensible extension of the fuzzy mathematical morphology; second, it is a promising example of the potential of the soft color morphology; and third, it implies that the common denominator of both tasks is extensive enough to face them with similar tools: curvilinear object detectors.

Resumen

La morfología matemática es un conjunto de técnicas de procesamiento de imagen en escala de grises. Se basa en dos operadores, la dilatación y la erosión, que respectivamente agrandan y disminuyen los objetos. En esta tesis, generalizamos estos operadores para procesar imágenes multivariadas, introduciendo así la dilatación suave en color y la erosión suave en color. Estos operadores están diseñados considerando espacios de color genéricos pero, al mismo tiempo, para procesar imágenes naturales de acuerdo con las nociones de *agrandar* y *disminuir* los objetos. Además de preservar los colores, otras propiedades teóricas son transferidas desde la morfología matemática borrosa. La dilatación y la erosión suaves en color pueden combinarse, tal y como se combinan la dilatación y erosión borrosas, para crear operadores con un comportamiento complejo. Se han diseñado muchas de estas combinaciones para afrontar tareas diversas, que pueden ser ahora utilizadas con imágenes en color: filtrado de ruido, corrección de contraste, segmentación de objetos o reconocimiento de formas, entre otras. En esta tesis, además, proponemos una definición de objetos curvilíneos para unificar el estado del arte: muchos problemas de procesamiento de imagen consideran la segmentación de objetos con forma tubular que se diferencian del fondo circundante. En particular, estudiamos dichos problemas para extraer su denominador común. Sintetizamos este estado del arte mediante la categorización tanto de las técnicas utilizadas para segmentar objetos curvilíneos como de las características de éstos que se consideran de interés. Además, diseñamos algoritmos basados en operadores morfológicos para segmentar objetos curvilíneos. Utilizamos la morfología matemática borrosa para segmentar vasos sanguíneos en fotografías del fondo del ojo y la morfología suave en color para detectar vello en imágenes dermoscópicas. Ambas morfologías consideran diferentes implementaciones de erosión y dilatación. Sin embargo, la dilatación y la erosión de cada morfología pueden ser combinadas de manera similar. Ambos algoritmos presentan unos resultados satisfactorios en comparación con otros trabajos publicados en la literatura científica. Esto tiene varias implicaciones: primero, la morfología suave en color es una extensión comprensible de la morfología matemática borrosa; segundo, constituye un ejemplo prometedor del potencial de la morfología suave en color; y tercero, implica que el denominador común de ambas tareas es suficientemente ámplio como para afrontarlas con herramientas similares: detectores de objetos curvilíneos.

Resum

La morfologia matemàtica és un conjunt de tècniques de processament d'imatge en escala de grisos. Es basa en dos operadors, la dilatació i l'erosió, que respectivament engrandeixen i disminueixen els objectes. En aquesta tesi, generalitzem aquests operadors per processar imatges multivariades, introduint així la dilatació suau en color i l'erosió suau en color. Aquests operadors estan dissenyats considerant espais de color genèrics però, al mateix temps, per processar imatges naturals d'acord amb les nocions d'*engrandir* i *disminuir* els objectes. A més de preservar els colors, altres propietats teòriques són transferides des de la morfologia matemàtica borrosa. La dilatació i l'erosió suaus en color es poden combinar, tal i com es combinen la dilatació i erosió borroses, per crear operadors amb un comportament complexe. S'han dissenyat moltes d'aquestes combinacions per afrontar diverses tasques, que poden ser ara utilitzades amb imatges en color: filtratge de renou, correcció de contrast, segmentació d'objectes o reconeixement de formes, entre altres. En aquesta tesi, també proposem una definició d'objectes curvilinis per unificar l'estat de l'art: molts problemes de processament d'imatge consideren la segmentació d'objectes de forma tubular que es diferencien del fons circumdant. En particular, estudiem aquests problemes per a extreure el seu denominador comú. Sintetitzem aquest estat de l'art mitjançant la categorització tant de les tècniques utilitzades per a segmentar objectes curvilinis com de les característiques d'aquests que es consideren d'interés. A més, dissenyem algoritmes basats en operadors morfològics per segmentar objectes curvilinis. Utilitzem la morfologia matemàtica borrosa per segmentar vasos sanguinis en fotografies del fons de l'ull i la morfologia suau en color per detectar pèls en imatges dermoscòpiques. Totes dues morfologies consideren diferents implementacions d'erosió i dilatació. No obstant això, la dilatació i l'erosió de cada morfologia poden ser combinades de manera similar. Els dos algoritmes presenten uns resultats satisfactoris en comparació amb altres treballs publicats en la literatura científica. Això té diverses implicacions: primer, la morfologia suau en color és una extensió comprensible de la morfologia matemàtica borrosa; segon, constitueix un exemple prometedor del potencial de la morfologia suau en color; i tercer, implica que el denominador comú de les dues tasques és prou ample com per afrontar-les amb eines similars: detectors d'objectes curvilinis.

Agraïments

Voldria donar les gràcies a totes les persones que m'han acompanyat al llarg d'aquesta aventura.

Gràcies als meus directors de tesi, pel seu suport constant i pel seu tracte fàcil i proper. La seva dedicació plena s'ha traduït en una feina de la que estic més que satisfet. També a tots els integrants del grup de recerca SCOPIA, en el que m'he sentit benvingut des del primer dia.

Gràcies a tots els companys amb els que he rigut durant el dia a dia. Una abraçada molt forta a tots els companys del laboratori de gràfics i visió per computador, i als companys del departament pel bon ambient que he gaudit. Gràcies també al grup de recerca *Intelligent Systems* de la Universitat de Groningen per la seva acollida.

Gràcies, especialment, a la meva família. Són ells els que m'han animat durant el camí i m'han recolzat en tot moment, especialment els dies més durs i feixucs. Els hi voldria dedicar aquesta tesi per la seva ajuda indispensable per arribar fins aquí.

Scientific Contributions and Funding

The majority of results presented in this thesis have been published in the scientific literature. In particular, the following journal articles have been developed based on the main contributions that this work represents.

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. Soft Color Morphology – A Fuzzy Approach for Multivariate Images. *Under review with major revision in the Journal of Imaging and Vision* (2018).

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. A real-time fuzzy morphological algorithm for retinal vessel segmentation. *Journal of Real-Time Image Processing* (2017). In Press.

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. General-purpose curvilinear object detection with fuzzy mathematical morphology. *Applied Soft Computing* 60 (2017), 655–669.

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. A Survey on Curvilinear Object Segmentation in Multiple Applications. *Pattern Recognition* 60 (2016), 949–970.

In addition to them, a number of works have been published in conference proceedings or as book chapters. They are listed as follows.

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. Skin Hair Removal in Dermoscopic Images Using Soft Color Morphology. *Conference on Artificial Intelligence in Medicine in Europe*. Springer. 2017, p. 322–326.

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. Soft color morphology. *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. 2017, p. 1–6.

– P. Bibiloni, M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. Fuzzy Black Top-Hat and Hit-or-Miss transformations and their applications. *Proceedings of Symposia on Mathematical Techniques Applied to Data Analysis and Processing (SMATAD)*. 2017, p. 37.

– P. Bibiloni, M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. Soft Computing Based Technique for Optic Disc and Cup Detection in Digital Fundus Images. *VipIMAGE 2017*. Volume 27. Lecture Notes in Computational Vision and Biomechanics. Springer, 2017, p. 82–91.

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. Detección de estructuras curvilíneas usando la transformación morfológica borrosa todo-nada. *XVIII Congreso Español sobre Tecnologías y Lógica Fuzzy. ESTYLF 2016*. EUSFLAT. 2016, p. 180–181.

– P. Bibiloni, M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. Mayor-Torrens t-norms in the Fuzzy Mathematical Morphology and Their Applications. *Fuzzy Logic and Information Fusion*. Springer, 2016, p. 201–235.

– P. Bibiloni, M. González-Hidalgo, and S. Massanet. Retinal Vessel Detection Based on Fuzzy Morphological Line Enhancement. *Advances in Artificial Intelligence*. Lecture Notes in Computer Science. Springer, 2015, p. 61–70.

xvi

- P. Bibiloni, M. González-Hidalgo, and S. Massanet. Vessel segmentation of retinal images with fuzzy morphology. *Computational Vision and Medical Image Processing V*. CRC Press. Taylor & Francis, 2015, p. 131–136.

In addition to the previous works, the PhD candidate has been involved in related projects that led to additional publications. Although they are not entirely covered in this thesis due to its scope, they represent an essential part of the candidate's training and research. They are the following conference works.

- M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. Fuzzy Hit-or-Miss Transform Using Uninorms. *Accepted in the 15th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2018)*. 2018.

- J. Molina, J. L. Chinchilla, P. Bibiloni, Á. Ríos, et al. Diseño de un protocolo para la adquisición de estudios PET con 18F-colina en modo dinámico con extracción simultánea de muestras sanguíneas arteriales. *36 Congreso Nacional de la Sociedad Española de Medicina Nuclear e Imagen Molecular*. 2017.

- S. Rubí, P. Bibiloni, M. Galmés, J. L. Chinchilla, et al. Cuantificación absoluta con muestreo arterial de la captación tumoral de 18F-fluorometilcolina por tomografía por emisión de positrones (PET) en pacientes con sospecha inicial de glioma de alto grado. *36 Congreso Nacional de la Sociedad Española de Medicina Nuclear e Imagen Molecular*. 2017.

- S. Rubí, P. Bibiloni, M. Galmés, M. Toscano, et al. PET Kinetic Modeling with Arterial Sampling of 18F-Choline Uptake in Patients with a Suspected Initial Diagnosis of High Grade Glioma. *European Journal of Nuclear Medicine and Molecular Imaging (EANM'17)*. Volume 44. Springer, 2017, p. 615.

- S. Rubí, M. Toscano, M. Valiente, J. Valera, et al. Modelització cinètica per PET amb presa de mostres arterials de la captació de 18F-colina en gliomes d'alt grau. *XXIX Jornades de la Societat Catalana de Medicina Nuclear i Imatge Molecular*. 2017.

The candidate has benefited from the fellowship FPI/1645/2014 of the *Conselleria d'Educació, Cultura i Universitats* of the *Govern de les Illes Balears* under an operational program co-financed by the European Social Fund. In addition to it, he has also been partially supported through his participation in the following projects:

- "Herramientas operacionales y métricas para el tratamiento de imágenes y la fusión de información", national project AEI/FEDER TIN2013-42795-P.

- "Técnicas de Soft Computing para el tratamiento de la incertidumbre en el procesamiento de imagenes", national project AEI/FEDER TIN2016-75404-P.

- "Modelización cinética a través de Tomografía de Emisión de Positrones (PET) con 18F-colina como biomarcador de neuroimagen para el diagnóstico y pronóstico de gliomas de alto grado", national project PI15/01653, funded by the *Insituto de Salud Carlos III*.

- "Asesoramiento en un proyecto de investigación para la evaluación de la uña con onicomicosis", private project funded by *Syntax for Science*.

- "Recerca i desenvolupament d'un nou mètode i eina de processament automàtic d'imatges en l'àmbit clínic", public project PROCOE/2/2017, funded by the Government of the Balearic Islands.

Finally, the candidate has disseminated part of his work in some seminars, including a presentation at the *Departament de Ciències Matemàtiques i Informàtica* from the *Universitat de les Illes Balears* and a presentation at the *Intelligent Systems* research group, from the *Univeristy of Groningen*.

A la meva família.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Artificial intelligence is a set of emerging technologies that are changing the world we live in. Although the term *artificial intelligence* was coined more than 50 years ago, in the Dartmouth Summer Conference of 1956, it is still emerging due to the fast evolution of science and technology. The best simulations of human intelligence have always been based on the most sophisticated techniques. Artificial intelligence is changing each and every aspect of our life: financial investment, information broadcasting, transportation and even recruiting are nowadays supported by algorithms.

Task automation is one of the most ambitious goals of artificial intelligence. Its development will potentially have an exceptionally high impact: monotonous and hazardous labours will no longer be required to be done with human's hands. In the last years, task automation has matured as a discipline in itself. For instance, computer algorithms are nowadays able to successfully classify images based on their content, write short news reports based on twitter comments or add color to grayscale photographs with outstanding results. These automation examples were out of reach just ten years ago. They add up to other milestones previously reached, such as the recognition of spoken digits, human-like automatic conversion via chat, autonomous car driving, the recognition of handwritten digits or defeating the world chess champion. More of these problems will become achievable as small, simple tasks are combined to provide more complex behaviour.

Image processing is playing a central role in this transformation. Let us emphasize that we receive vasts amounts of information through the sight, and we greatly depend on it. Due to recent developments, image processing has reclaimed its position as a major topic in the field of artificial intelligence with regard to its value and development pace.

Despite the popularity of task automation in some fields, medicine is still reticent to adopt it. Although this field typically adopts the latest technological developments, the central role of the physician is indisputable. For instance, specially designed devices may be used to acquire and reconstruct three-dimensional images which are, at the very end, manually analized to provide a diagnosis. In this context, technology is used as a tool rather than as an automation mechanism. In other words, artical intelligence is endorsed if, instead of deciding, it just helps.

The decision of a medical practitioner is undeniably preferred to automatic algorithms. The development and deployment of artical intelligence-based systems encounters many barriers. From a social point of view, they are not always accepted, as demonstrates the recent trend of creating robots *less* similar to humans to make them clearly different to people. A more philosophic criticism argues on the impossibility of modelling intelligence as an information processing system, or on the implicit assumption that we can internally represent all the information we acquire about the world [120]. Regarding task automation in medicine, there is a rationale that outweights the rest of them: the risk of being wrong. The consequences of a bad diagnosis are potentially very harmful. Both the patient and the physician prefer the latter's justified decision than that of a system whose internal working mechanism may be opaque.

Medical data typically contains incomplete, vague or even contradictory information. This poses as another factor that discourages the use of artical intelligence. Each patient has a different set

of characteristics, and therefore it is very hard to standardize them to apply a series of sequential rules to make a decision. For instance, some images are corregistered with standard models, known as atlas, to locate macroscopic regions. This corregistration may be inaccurate or even completely flawed whenever the patient had one anatomic region surgically removed. Besides, each device that gathers patient information is affected by one or more different types of noise. Three-dimensional volumetric images are reconstructed from physical measurements, histological diagnosis may be wrong with a non-negligible probability, and so on. In addition to the above, a myriad of pathologies are responsible for even more variability. Therefore, the unreliability of biomedical imagery may have a considerable impact in the outcome of automatic processing techniques.

Fuzzy logic is a tool specifically developed to deal with uncertainty and vagueness [198]. It is an attempt to formalize knowledge while admitting that information may not be known in its entirety. Fuzzy logic is a many-valued logic, in which propositions are not simply *true* or *false*, but they are assigned a degree of truthfulness ranging from 0 to 1. In contrast to probability theory, in which all events have a *known* numeric value of its likelihood, fuzzy logic may be used to model ignorance: for instance, assigning a low truth value to both an event and its negation. In addition to incomplete information, it can also model vagueness, contradiction and, of course, certainty.

Some computer vision techniques have been designed taking advantage of the underlying principles of fuzzy logic [37]. They do so to deal with noisy images or to represent the information they contain. In a grayscale image, each pixel is assigned a value in the range $\{0, 1, \ldots, 2^{n-1}\}$, typically with n being 8, 16 or 24. However, from a theoretical point of view, they can be rescaled into the interval $[0, 1]$. Some authors observe that this assignment falls into the definition of fuzzy set, and interpret the value of each pixel as the degree of it representing a bright object. Following the same line, the operations used with fuzzy logic values, such as fuzzy conjunctions or fuzzy disjunctions, are used to process each pixel's value. Also, other authors consider uncertainty when encoding the information represented in the image. Vague statements may be extracted from a noisy image, such as *one object is above some other object*; *one object is big*; or *one object is convex*. These vague statements may be valid up to a certain point.

Fuzzy mathematical morphology is a computer vision tool which implements elements from fuzzy logic [55]. It processes grayscale images based on two operations, the erosion and the dilation. The former shrinks bright objects, whereas the latter enlarges them. Both operations provide a modified version of the original image, and can be combined to design systems with a complex behaviour. They can be combined either sequentially or iteratively, until a stop criteria is met.

In the literature, the same techniques ara adapted once and again to solve similar tasks. To solve a specific problem (*e.g.* vessel segmentation), we usually split it into several simple tasks (*e.g.* red spot detection), and then adapt a specific technique to address each task (*e.g.* linear filtering). Nowadays, practitioners reuse techniques: instead of inventing brand new algorithms, the most effective ones are adapted to solve different tasks. In contrast, task reusing is much less common. Although different problems sometimes involve solving a similar task, all of them are faced independently, and solved following similar paths accross different fields.

Curvilinear object detection is an example of a task found in many computer vision applications [69, 113, 92]. Curvilinear objects are thin, elongated regions, clearly different from their background. They can have self-intersections or form a complex network. A great variety of objects can be thought of as curvilinear objects: vessels in certain medical images, roads and rivers from aerial images, road surface marking, cracks in industrial product manufacturing, etc. Considering the common denominator of all such tasks is of paramount importance to keep standing on the shoulders of giants.

1.1 Objectives of this Thesis

The motivation of this thesis arises from pursuing the automation of tasks using tools from the field of image processing. More specifically, we focus on its applications to medical imagery and

its mechanisms to deal with imprecise information. The thesis is aimed at fulfilling the following milestones:

Objective 1. Extend the fuzzy mathematical morphology to multivariate images.

The operators of the fuzzy mathematical morphology are capable of dealing with grayscale images. They are, as a matter of fact, an extension of the original morphology operators, envised to deal with binary images. Being able to process multivariate images, like natural color images or corregistered PET/CT scans, would allow practitioners to leverage their knowledge of these operators in a wider range of situations.

There is no successful definition of color morphology due to the difficulty of interpreting which objects must be enlarged (or shrinked) by the dilation (or the erosion). Bright pixels in grayscale or binary images are assumed to represent objects. In both cases we can decide which pixels have priority. Regarding color images, however, there is no such priority: different colors become incomparable. We emphasize that, in order to modify the objects, we must first differentiate them from their surrounding background.

Objective 2. Design automatic algorithms for certain types of medical imagery taking into account the uncertainty they contain.

This second milestone is aimed at applying the fuzzy and color mathematical operators to real problems. It is aimed at studying their efficiency, their performance and, most importantly, their behaviour.

The purpose of extracting information from medical images is aiding the phyisician or the researcher. This aid may have the form of automating a manually tedious task, preprocessing images to uniformize or clean them or, in its highest form, exposing correlations between features of the images and the pathologies they represent. We are aware of the great complexity this may pose, but even designing and testing a seemingly simple task may be challenging enough to test whether the techniques used are appropriate or not.

Specifically, we consider two types of medical images:

(a) Fundus photographies. They are images of the retina, located at the back of the ocular globe. The retina is a relatively visible region in which thin capillaries can be observed. This may be one of the first places in which the symptoms of some specific pathologies are revealed, such as in the case of diabetes. Although they are taken with standard cameras in the RGB format, we only use their green channel. Thus, we study them with fuzzy mathematical morphology operators.

(b) Dermoscopic images. Images of the skin taken with a specific device, the dermoscope. They are magnified photographs of a small region of the skin taken to study a single lesion. The dermoscope increases the accuracy of the diagnosis of, among others, malignant melanoma. It produces natural color images and can be processed with multivariate operators.

In both cases, we are interested in the underlying vagueness and uncertainty. First, we must study the images: the types of noise they have, the anatomic structures and pathologies they may contain, and what represents each pixel's value. Second, the techniques employed must be adapted to tackle each of the aforementioned sources of imprecision: it is important to know how each specific imprecision is being addressed.

Objective 3. Working towards the use of task automation systems in real settings.

This thesis is motivated by a real world problem and, thus, has sense provided it is

eventually transferred to the real situation in which it can have a salient impact. Even though algorithms are designed by computer scientists, they are normally leveraged by professionals from other fields. Thus, we must consider the constraints of a multi-disciplinar project: professionals from different fields must work towards a common goal.

A first route to consolidate knowledge for technology transfering is task reusing. Besides splitting complex problems into several simple tasks, they must be solved just once. Building on previous discoveries is, obviously, much easier than reinventing the wheel many times.

Second, given the specific field of application of this thesis, a close interaction with physicians is required. As previously discussed, introducing novelties in the clinical practice is difficult due to legislation, adoption by practitioners and the time required to employ it. Joint work between computer scientists and physicians is essential to understand their needs and the problems they face.

1.2   Contributions

This thesis is organized from general-purpose concepts to more specific ones. Chapter 2 introduces the common notation used throughout the thesis and some general concepts in computer vision for the sake of completeness. The rest of the thesis is organized in several chapters, being each of them a separate contribution. In the following, we enumerate them.

Chapter 3 is an extensive review of the task of curvilinear object detection. We found that curvilinear objects are present in a variety of images, including retinal images (see Ch. 5). First, we study which applications consider curvilinear objects. More importantly, we study how they model them and the disturbances they may present. Afterwards, we study and compare more than a hundred of independent techniques to segment such objects. They are classified according to the underlying mathematical technique employed to characterize curvilinear objects. This chapter constitutes a contribution towards fulfilling Objective 3 by synthesizing the state-of-the-art knowledge in computer vision with regard to segmenting curvilinear objects.

In Chapter 4, we introduce two curvilinear object detectors based on fuzzy mathematical morphology. After studying the most successful models of curvilinear objects, we propose two fast and generic algorithms to segment them. They are thoroughly examined to provide the practitioners with as much information as possible regarding their usage. Also, they are compared with other general-purpose curvilinear object detectors, showing the strengths and weaknesses of all the alternatives. Again, aiming at task reusing, this contribution chases Objective 3.

Chapter 5 introduces algorithms to segment vessels in eye-fundus images. To do so, we first consider the different methods studied in Ch. 3. After a comparison, we develop an algorithm based on one of the curvilinear detectors introduced previously. We analyse its behaviour showing that it is both fast and accurate. This contribution pursues Objective 2: it is implemented using operators from the fuzzy mathematical morphology, taking into account the imprecision of pixel values and the definition of vessels in this context, which is vague due to its subjective nature.

In Chapter 6 we introduce the operators of the soft color morphology, our proposal for extending mathematical morphology to multivariate images. Based on the fuzzy mathematical morphology operators, we define both a dilation and an erosion operator to process color images. In particular, we transfer to this setting the same mechanisms to deal with uncertainty: both encoded in the pixels values and by considering non-binary structuring elements that can be used to express loose concepts such as *thin* or *to the right of*. This chapter represents a contribution that, to the extent of our ambition, completely attains Objective 1.

Chapter 7 is an application of the aforementioned color morphological operators to process dermoscopic images. In particular, we propose a method to process them by removing hair and other

artifacts that hinder the automatic lesion segmentation. This contribution is mainly focused on Objective 2. Also, the content in this chapter has been developed taking into account the features of dermoscopic images and the information dermatologists obtain from them, pursuing also Objective 3.

We conclude this thesis with a series of lines of research to extend and apply the results obtained with it. In particular, we list several interesting problems that have arised during the study that this thesis represents. By analysing the contributions of this thesis from a higher perspective, we are at a better position to detect its major strengths and weaknesses.

Chapter 2

Preliminaries

In this chapter, we introduce all the required concepts that are used throughout this thesis. It includes notation and concepts common to all chapters, and also techniques already developed that are tightly related to the contributions of the thesis.

In Sect. 2.1, we introduce the mathematical notation to express colors, channels and images. A particular case, the CIELab color space, is introduced in Sect. 2.2. In Sect. 2.3 and 2.4 we introduce the fuzzy logic operators and the fuzzy mathematical morphology operators, which are based on the former. Finally, Sect. 2.5 present the hysteresis and the CLAHE algorithm, two well-known computer vision algorithms.

2.1 Formal Description of Images

To mathematically describe algorithms that deal with colors, channels and images, we first describe such concepts with a formal language.

Definition 2.1. — *We define the following elements.*

- *A* channel, $\mathscr{C}$*, is a set of real values,* $\mathscr{C} \subset \mathbb{R}$*.*

- *A* color space, $\mathfrak{C}$*, is the cartesian product of a series of channels* $\mathfrak{C} = \mathscr{C}_1 \times \ldots \times \mathscr{C}_m$*.*

- *A* $\mathfrak{C}$*-encoded color is one element of the color space,* $c \in \mathfrak{C}$*.*

- *A* $\mathfrak{C}$*-encoded multivariate image or simply a* $\mathfrak{C}$*-encoded image, A, is a map* $A : \mathbb{Z}^n \to \mathfrak{C}$*, where n is its* spatial dimension *and* $\mathfrak{C}$ *is its color space.*

- *Any image A is associated to its* support $d_A \subset \mathbb{Z}^n$*, the spatial region where the relevant information is found. That is, we will be only interested in the values* $A(x)$ *for* $x \in d_A$*.*

Some remarks must be mentioned. First, without loss of generalization, we will always consider that the first channel is $\mathscr{C}_1 = [0, 1]$. Otherwise, since $\mathscr{C}_1 \subset \mathbb{R}$, we can map it to $[0, 1]$ with a monotonic bijection—for instance, with the sigmoid function. Second, a *grayscale image* is a particular case of multivariate image, which has only one channel. In this case, $m = 1$ and, therefore, its color space is $\mathfrak{C} = [0, 1]$. Finally, the support of an image A is simply a region within the spatial location $\mathbb{Z}^n$ that indicates which part of the image is of interest. It can be assumed to contain no meaningful information outside it.

We employ structuring elements whose values range in $[0, 1]$ to compute the morphological operators that will be used in this thesis. We also define operations to deal with movements in the spatial domain.

Definition 2.2. — *We define the following elements.*

- *A* structuring element, *B, is a grayscale image* $B : \mathbb{Z}^n \to [0, 1]$*, where n is its* spatial dimension*.*

- *The* reflection *of a structuring element B is the structuring element* $\bar{B}$ *defined by* $\bar{B}(x) = B(-x)$*.*

- *The* spatial translation *by a vector $v \in \mathbb{Z}^n$, T_v, is a map from subsets of $\mathbb{Z}^n$ to subsets of $\mathbb{Z}^n$ such that*

$$\forall d \subset \mathbb{Z}^n, \ \forall a \in \mathbb{Z}^n, \quad a \in T_v(d) \iff a - v \in d.$$

Similarly to multivariate images, a structuring element B is always associated to its *support* $d_B \subset \mathbb{Z}^n$. We will consider of interest the values of the structuring element B only within its domain d_B. We will always consider structuring elements with the same spatial dimension n as the images we are processing. Typically, but not always, structuring elements have a value of one in the origin: $B(0) = 1$.

2.2   The CIELab Color Space

CIELab, or CIE 1976 L* a* b*, is a color space used to represent images in the visible spectrum. It has three channels. The first one, L*, matches the human perception of lightness, whereas the other two, a* and b*, conceil its chromatic information. Besides, it is perceptually uniform: the Euclidean distance between two colors is proportional to their difference when perceived by humans [190].

Typical implementations of the CIELab color space consider the first channel to be L* $= [0, 100]$. Throughout this thesis, we divide such lightness values by 100 so that our first channel is exactly $[0, 1]$. The two chromatic components, a* and b*, are roughly in the range $[-110, 110]$, although not all combinations yield valid RGB colors. In Fig. 2.1 all RGB colors have been represented according to their CIELab representation.
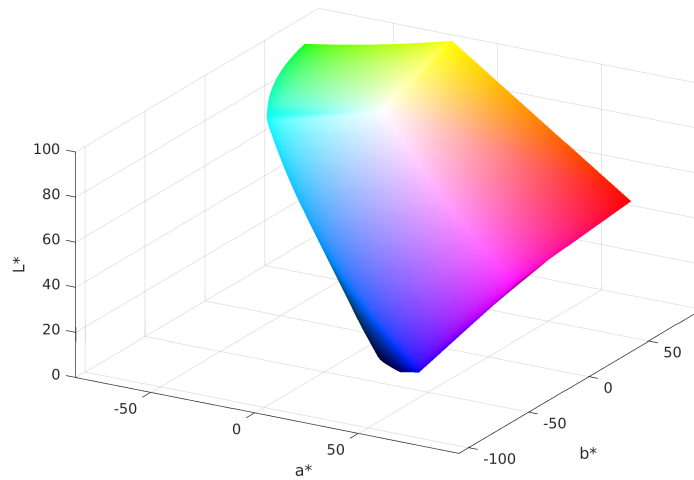
FIGURE 2.1: CIELab gamut obtained by representing all RGB colors in the CIELab color space.

Conversion between CIELab and grayscale is straightforward, as shown in the following definition.

Definition 2.3. (Conversion of CIELab to and from grayscale [190]) — *The map π, that converts a CIELab-encoded color (l, a, b) into a grayscale color, is defined as:*

$$\pi : \begin{array}{ccc} L^*a^*b^* & \longrightarrow & [0, 1] \\ (l, a, b) & \longmapsto & l. \end{array}$$

The map ɪ, that converts a grayscale color G into a CIELab-encoded color, is defined as:

$$
\begin{aligned}
\iota: \quad [0,1] &\longrightarrow \quad L^*a^*b^* \\
G &\longmapsto \quad (G,0,0).
\end{aligned}
$$

That is, simply by reducing the CIELab-encoded image into the L^* channel (*i.e.* forgetting the channels a^* and b^*) provides a grayscale version. Whenever non-grayscale colors are considered, the conversion between RGB and CIELab is complex in order to achieve the perceptual invariance of the CIELab color space. For more information regarding the CIELab color space and its relation to other color spaces such as RGB and XYZ, see Wyszecki and Stiles [170, 190].

2.3 Fuzzy Logic Operators

The fuzzy morphological operators are based on the use of fuzzy operators, which play the role in fuzzy logic of the corresponding binary operators in classical logic. They extend the binary notions to a multi-valued setting, considering that their input—originally thought of as a truth value—can take intermediate values between absolute truth and absolute falsehood. We briefly introduce them here due to their use in the Fuzzy Mathematical Morphology, in Sect. 2.4, and in its extension to multivariate images, the Soft Color Morphology, in Ch. 6, which is one of the main contributions of this thesis. More information about aggregation functions can be found in [27] and about fuzzy implication functions in [13, 12].

We start by defining a well-known general family of functions.

Definition 2.4. — *A n-ary aggregation function, with $n > 1$, is a function of the form $f : [0,1]^n \to [0,1]$ such that:*

- *$f(0,0,\ldots,0) = 0$ and $f(1,1,\ldots,1) = 1$, and*

- *for all $\mathbf{x}, \mathbf{y} \in [0,1]^n$ such that $\mathbf{x} \leqslant \mathbf{y}$, then $f(\mathbf{x}) \leqslant f(\mathbf{y})$,*

where $\mathbf{x} \leqslant \mathbf{y}$ is true if it holds for every component.

Next, we recall the definitions of the binary aggregation functions that have a conjunctive-like behaviour.

Definition 2.5. — *A fuzzy conjunction, C, is a bivariate aggregation function in which $C(1,1) = 1$ and $C(1,0) = C(0,1) = 0$.*

Definition 2.6. — *A semi-norm is a fuzzy conjunction C with neutral element 1. That is, for all $x \in [0,1]$, $C(1,x) = C(x,1) = x$.*

Definition 2.7. — *A t-norm is a semi-norm that is associative and commutative.*

Similarly, we also define the binary aggregation functions that present a disjunctive-like behaviour:

Definition 2.8. — *A fuzzy disjunction D is a bivariate aggregation function such that $D(0,0) = 0$ and $D(1,0) = D(0,1) = 1$.*

Definition 2.9. — *A t-conorm S is a fuzzy disjunction that is associative, commutative and fulfills $S(0,x) = x$ for all $x \in [0,1]$.*

Another operator that plays an important role in both fuzzy and classical logic is the negation.

Definition 2.10. — *A strong fuzzy negation is a decreasing map $N : [0,1] \to [0,1]$ that satisfies $N(0) = 1$ and $N(1) = 0$, and that is involutive: $N(N(x)) = x$ for all $x \in [0,1]$.*

Given a t-norm and a strong fuzzy negation, we can define a t-conorm as follows.

Proposition 2.11. — *Let T be a t-norm, let N be a strong fuzzy negation. Then, the operator $S_{T,N}$, defined as $S_{T,N}(x,y) = N(T(N(x), N(y)))$, is a t-conorm.*

The operator above defined is called the *dual t-conorm of T with respect to N*.

Lastly, we recall the definition of a fuzzy implication function. It is the operator that plays the role, in fuzzy logic, of the binary implication.

Definition 2.12. — *A fuzzy implication function is a map $I : [0,1] \times [0,1] \to [0,1]$ that is decreasing in the first variable and increasing in the second, and that satisfies $I(1,0) = 0$ and $I(0,0) = I(1,1) = 1$.*

A multitude of families of fuzzy implication functions have been defined in the literature [13, 12]. One of the most used ones is defined from a conjunction by residuation.

Proposition 2.13. — *Let C be a fuzzy conjunction that satisfies $C(1, x) > 0$ for all $x > 0$. Then, its residual operator, defined as $I(x, y) = \sup \{t \in [0,1] \mid C(x,t) \leqslant y\}$ for all $x, y \in [0,1]$, is a fuzzy implication function.*

In particular, we say that I is the residual implication (or R-implication) of C.

We also state some properties that are used throughout this article [143].

Property 2.14. — *Let f be a bivariate aggregation function and let N be a strong fuzzy negation. Then:*

(NC) *f and N fulfill the Non-Contradiction principle whenever $f(x, N(x)) = 0$ for all $x \in [0,1]$.*

Property 2.15. — *Let I be a fuzzy implication function. Then:*

(NP) *I fulfills the left neutrality principle whenever $I(1, y) = y$ for all $y \in [0,1]$.*

(OP) *I fulfills the ordering property whenever $x \leqslant y \Leftrightarrow I(x, y) = 1$ for all $x, y \in [0,1]$.*

Finally, we introduce the notion of adjunction, which relates the monotonicity of two operators. Due to its nature, this notion is useful to prove the idempotence of certain fuzzy morphological operators [167]. We introduce this notion for sets whose elements have a weak ordering relation that is reflexive, antisymmetric and transitive—but does not need to be total.

Property 2.16. — *Les $\mathcal{X}$ be a partially ordered set, and let $f, g : \mathcal{X} \to \mathcal{X}$ be two mappings from the set to itself. Then, f and g form an adjunction pair if for all $x_1, x_2 \in \mathcal{X}$,*

$$f(x_1) \leqslant x_2 \iff x_1 \leqslant g(x_2).$$

In particular, semi-norms and implications form adjunction pairs with respect to the totally ordered interval $[0,1]$ under certain constraints. This is reflected in the following proposition.

Proposition 2.17. (Adjunction pair [12]) — *Let C be a left-continuous semi-norm and I the R-implication of C. Then, for all $x \in [0,1]$, the functions $C(x, \cdot)$ and $I(x, \cdot)$ form an adjunction pair. That is, for all $y_1, y_2 \in [0,1]$,*

$$C(x, y_1) \leqslant y_2 \iff y_1 \leqslant I(x, y_2).$$

Throughout this thesis we use the operators that appear in Tab. 2.1 jointly with the standard negation (*i.e.* $N(x) = 1 - x$). They are the pairs of t-norms and their corresponding R-implications: the minimum t-norm $T_{\mathbf{M}}$ and the Gödel implication $I_{\mathbf{GD}}$, the product t-norm $T_{\mathbf{P}}$ and the Goguen implication $I_{\mathbf{GG}}$, and the Łukasiewicz t-norm $T_{\mathbf{L}}$ jointly with the Łukasiewicz implication $I_{\mathbf{L}}$. In Tab. 2.1, (NC)* denotes the Non-Contradiction principle with respect to the standard negation. Also, to express piecewise functions, we use $\mathbb{1}_\phi$, that equals to 1 whenever ϕ is true, and to 0 otherwise. We remark that all R-implications fulfill (NP).

TABLE 2.1: Each row contains a t-norm, its dual t-conorm with respect to $N(x) = 1 - x$, and its R-implication.

t-norms	(NC)*
$T_{\mathbf{M}}(x,y) = \min(x,y)$	✗
$T_{\mathbf{nM}}(x,y) = \min(x,y) \cdot \mathbb{1}_{x+y>1}$	✓
$T_{\mathbf{P}}(x,y) = x \cdot y$	✗
$T_{\mathbf{L}}(x,y) = \max(0, x+y-1)$	✓

t-conorms
$S_{\mathbf{M}}(x,y) = \max(x,y)$
$S_{\mathbf{nM}}(x,y) = \max(x,y) \cdot \mathbb{1}_{x+y<1} + \mathbb{1}_{x+y\geqslant1}$
$S_{\mathbf{P}}(x,y) = x + y - xy$
$S_{\mathbf{L}}(x,y) = \min(1, x+y)$

Fuzzy implication functions	(NP)	(OP)
$I_{\mathbf{GD}}(x,y) = \mathbb{1}_{x\leqslant y} + y \cdot \mathbb{1}_{x>y}$	✓	✓
$I_{\mathbf{FD}}(x,y) = \mathbb{1}_{x\leqslant y} + \max(1-x,y) \cdot \mathbb{1}_{x>y}$	✓	✓
$I_{\mathbf{GG}}(x,y) = \min(1, y/x)$	✓	✓
$I_{\mathbf{L}}(x,y) = \min(1, 1-x+y)$	✓	✓

2.4   Fuzzy Mathematical Morphology

Mathematical morphology, introduced for binary images by by Serra and Matheron [79], embraces a set of techniques to process images. It is based on two simple and local operations: the erosion and the dilation. The former shrinks objects whereas the latter enlarges them, according to a specific template called *structuring element*. They can be combined sequentially or iteratively (*i.e.* until a halt condition is reached) to design elaborated image processing operators. Mathematical morphology is the set of these image processing techniques, including from simple operators to very complex ones. Mathematical morphology was originally designed to deal with binary images. In them, pixels having a value of 1 are interpreted as representing an object, whereas the background is represented with the value 0. For more information on binary mathematical morphology, see [157], and [158] for the lattice-based definition of erosion and dilation.

Fuzzy mathematical morphology [129, 55, 38] is an extension of binary mathematical morphology that processes grayscale images with grayscale structuring elements. It considers that grayscale values can be interpreted as fuzzy values, with which several types of imprecision in images can be represented, such as imprecision in spatial location or in the membership of an object to a class [36]. The structuring element, of the same nature, is also represented as a fuzzy set. In particular, each pixel's value can represent the degree with which the pixel represents a specific entity. For instance, in Ch. 3, we consider curvilinear objects as thin, elongated objects whose photometric features are different from the ones of its neighbourhood. The geometric information, *thin* and *elongated*, is encoded in the shape of the structuring element. The photometric information, on the other hand, is encoded in the pixels' values.

Grayscale images, thus, are interpreted as a fuzzy sets [36, 167]. The fuzzy mathematical erosion reflects the degree with which the (shifted) structuring element *fits* into the image. On the other hand, the fuzzy mathematical dilation reflects the degree of similarity between the image and the (shifted) structuring element.

Fuzzy mathematical morphology is based on two operators, the fuzzy erosion and the fuzzy dilation. They are introduced in Defs. 2.18 and 2.19.

Definition 2.18. — *Let I be a fuzzy implication function and let A, B be grayscale images. Then, the fuzzy erosion of A by B, $\mathcal{E}_I(A,B)$, is defined as:*

$$\mathcal{E}_I(A,B)(y) = \inf_{x \in d_A \cap T_y(d_B)} I\big(B(x-y), A(x)\big).$$

Definition 2.19. — *Let C be a fuzzy conjunction and let A, B be grayscale images. Then, the fuzzy dilation of A by B, $\mathcal{D}_C(A, B)$, is defined as:*

$$\mathcal{D}_C(A, B)(y) = \sup_{x \in d_A \cap T_y(d_B)} C\big(B(x-y), A(x)\big).$$

As in the binary case, the dilation enlarges objects—regions with higher grayscale values than their surroundings—, while the erosion shrinks them. We recall that d_A denotes the support of the image A, as introduced in Def. 2.1.

Proposition 2.20 extends the adjunction proposition from fuzzy operators (see Prop. 2.17). In particular, fuzzy operators that form an adjunction pair in $[0, 1]$ may be used to implement a dilation operator and an erosion operator that form an adjunction pair with respect to the images they process. We recall that Property 2.16 is a monotonicity property on a partially ordered set that relates two mappings. In the case of grayscale images, the monotonicity is given in terms of image inclusion: given two grayscale images A_1, A_2 with domain d, we say that A_1 is *lower* than A_2, denoted as $A_1 \subset A_2$, if each corresponding pixel has a lower value. That is, $A_1 \subset A_2$ if and only if for all $x \in d$, $A_1(x) \leqslant A_2(x)$.

Proposition 2.20. (Adjunction pair [129]) — *Let C be a left-continuous semi-norm and I the R-implication of C. Let B be a structuring element. Then, the operators $\mathcal{D}_C(\cdot, \overline{B})$ and the $\mathcal{E}_I(\cdot, B)$ are an adjunction pair. That is, for all grayscale images A_1, A_2,*

$$\mathcal{D}_C(A_1, \overline{B}) \subset A_2 \iff A_1 \subset \mathcal{E}_I(A_2, B).$$

In the framework of De Baets [55] the fuzzy dilation and the fuzzy erosion do not always form an adjunction pair since a general fuzzy conjunction and a general fuzzy implication function are used to define the fuzzy morphological operators. On the other hand, as stated in Prop. 2.20, when C is a left-continuous t-norm and I is its R-implication, the fuzzy erosion and the fuzzy dilation form an adjunction pair. This is a desirable relation due to the theoretical properties that hold when the erosion and dilation form an adjunction pair [167].

The fuzzy erosion and fuzzy dilation can be combined to design a wide variety of transformations. Two well-known examples of such combinations are the fuzzy opening and the fuzzy closing, introduced in Defs. 2.21 and 2.22 as in the binary case [129].

Definition 2.21. — *Let C be a fuzzy conjunction, let I be a fuzzy implication function and let A, B be grayscale images. Then, the fuzzy opening, $\mathcal{O}_{C,I}(A, B)$, is defined as:*

$$\mathcal{O}_{C,I}(A, B) = \mathcal{D}_C(\mathcal{E}_I(A, B), \overline{B}).$$

Definition 2.22. — *Let C be a fuzzy conjunction, let I be a fuzzy implication function and let A, B be grayscale images. Then, the fuzzy closing, $\mathcal{C}_{C,I}(A, B)$, is defined as:*

$$\mathcal{C}_{C,I}(A, B) = \mathcal{E}_I(\mathcal{D}_C(A, B), \overline{B}).$$

Although they have an opposite behaviour, the fuzzy erosion and fuzzy dilation are not inverse of each other. Similarly, the fuzzy opening and the fuzzy closing also have an opposite behaviour given their definition. The fuzzy opening strongly affects small and bright objects, being similar to the original elsewhere. Such objects completely disappear during the erosion and, thus, can not be recovered by the dilation. The fuzzy closing has a similar behaviour with respect to small and dark objects. Jointly with the fuzzy erosion and fuzzy dilation, they are monotonic operators when using appropriate parameters, as is shown in Proposition. 2.23 [129, Prop. 45].

Proposition 2.23. (Extensiveness of Fuzzy Morphological Operators) — *Let C be a semi-norm, let I be an fuzzy implication function that satisfies* (NP). *Let A, B be grayscale images such that $B(0) = 1$.*

Then,

$$\mathcal{E}_I(A,B) \subset A \subset \mathcal{D}_C(A,B).$$

If, in addition to it, I is the R-implication of C, then,

$$\mathcal{E}_I(A,B) \subset \mathcal{O}_{C,I}(A,B) \subset A \subset \mathcal{C}_{C,I}(A,B) \subset \mathcal{D}_C(A,B).$$

We remark that all *R*-implications *I* of a given semi-norm fulfill the border condition (NP).

Another well-known morphological operation is the fuzzy white/black top-hat operator, introduced in Def. 2.24. It captures the dissimilarity between the fuzzy opening/closing and the original image.

Definition 2.24. — *Let C be a left-continuous semi-norm and let I be its residual implication. Let A, B be grayscale images such that $B(0) = 1$. Then, the fuzzy black top-hat, also known as fuzzy top-hat by closing, $\mathcal{BTH}_{C,I}(A,B)$; and the fuzzy white top-hat, also known as fuzzy top-hat by opening, $\mathcal{WTH}_{C,I}(A,B)$, are defined as:*

$$\mathcal{WTH}_{C,I}(A,B) = A - \mathcal{O}_{C,I}(A,B), \quad \mathcal{BTH}_{C,I}(A,B) = \mathcal{C}_{C,I}(A,B) - A.$$

Finally, the fuzzy hit-or-miss is a more complex operator that aims at locating specific shapes. It employs two structuring elements to enforce (i) the appearance of a specific shape and (ii) the absence of another specific shape. As in [71], it is defined as follows.

Definition 2.25. — *Let C be a fuzzy conjunction, let I be a fuzzy implication function, let N be a strong fuzzy negation and let A, B_1, B_2, be grayscale images. Then, we define the fuzzy hit-or-miss operator, $\mathcal{FHM}_{C,I,N}(A,B_1,B_2)$, is defined as:*

$$\mathcal{FHM}_{C,I,N}(A,B_1,B_2) = C\big(\mathcal{E}_I(A,B_1),\, \mathcal{E}_I(N(A),B_2)\big).$$

The first term, $\mathcal{E}_I(A,B_1)$, measures the inclusion of B_1 in the image *A*, whereas the second one, $\mathcal{E}_I(N(A),B_2)$, measures the exclusion of B_2 in *A*. Thus, the fuzzy hit-or-miss enhances the areas that are, at the same time, fitted by B_1 and excluded from B_2.

It generalizes the binary hit-or-miss transform while admitting grayscale input images and grayscale structuring elements. It is invariant to translations and, whenever $B_2 = N(B_1)$ and the implication *I* fulfills (OP), it outputs 1s only in the locations of *A* in which it appears exactly B_1 [71].

2.5 Other Computer Vision Algorithms

Throughout this thesis, some other computer vision algorithms are used. For the sake of completeness, we describe their purpose and behaviour.

2.5.1 Hysteresis

Hysteresis is a well-known procedure to convert a grayscale image into a binary one. It is more sophisticated than a simple threshold, since it also employs information about the connectivity of the pixels. Hysteresis uses two thresholds, $t_{\text{low}}, t_{\text{high}} \in [0,1]$, such that $t_{\text{low}} \leqslant t_{\text{high}}$. It works as follows: a pixel is classified as "positive" as long as it is placed in a connected region of pixels greater or equal than t_{low} such that at least one of them is also greater or equal than t_{high} [122].

This procedure benefits pixels arranged in connected components, since a pixel may be selected or not depending on its situation with respect to the other ones. Besides being fast, this operation does not consider just a fixed-sized neighbourhood, since a pixel may be in the same connected component that other very distant one. Throughout this thesis, we consider 8-connectivity pixels due to the possible appearance of 1-pixel width objects whose obliquous direction would not be captured by using the 4-connectivity.

2.5.2   Contrast Limited Adaptive Histogram Equalization

Contrast Limited Adaptive Histogram Equalization (CLAHE), introduced by Pizer *et al.* [142], is an algorithm to visually improve grayscale images, modifying them in order to equalize their histogram. It re-maps the grayscale values of images so that their pixel values are more equidistributed. Originally designed to deal with medical images, CLAHE is a fast algorithm that avoids noise amplification in uniform regions.

The *contrast limited* behaviour is related to limiting the maximum slope of the re-mapping function. This attempts to avoid amplifying noise in uniform regions by enforcing that two similar pixels do not end up with very distinct values. It is controlled by a parameter called *clip limit*, whose behaviour is shown in Fig. 2.2.
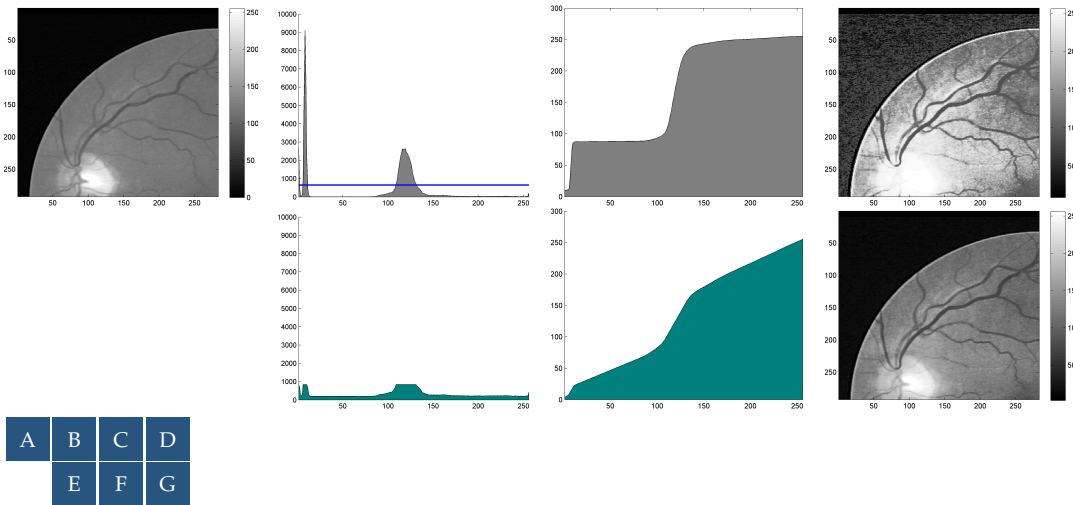
FIGURE 2.2: Histogram equalization with and without contrast limit: (A) original image, (B) histogram and clip limit, (C) cummulative histogram, (D) re-mapping of (A) according to (C), (E) clipped histogram, (F) cummulative clipped histogram, and (G) re-mapping of (A) according to (F).

Without applying the contrast limit the re-mapping function would be proportional to the cummulative distribution of the histogram within the region (see Fig. 2.2(D)), since this transformation would redistribute pixels among all grayscale levels equally. However, the cummulative distribution of an arbitrary histogram may present very high slopes. To avoid them, we crop the histogram at a value proportional to the clip limit, so the slope of cummulative histogram (and thus of the re-mapping function) is limited (Fig. 2.2(E)). The values of the histogram that are cropped are equally redistributed among all grayscale levels as a way to smooth and regularize the re-mapping function [142].

The *adaptive* behaviour is responsible of properly handling regions with different gray distributions. We remark that the visual example depicted in Fig. 2.2 is not an adaptive equalization, since just a single region was considered. To perform an adaptive equalization, the image is divided in different regions and different re-mapping functions are computed. Bilinear interpolation is used afterwards to avoid a blocking effect. That is, for each pixel we compute the re-mapping according to each of its surrouding regions, and the final result is a weighted interpolation of these values [204]. Specifically, let A be a gray-level image and (x, y) a pixel located among the centers of the regions $C_1 = (\alpha_1, \beta_1)$, $C_2 = (\alpha_2, \beta_2)$, $C_3 = (\alpha_3, \beta_3)$ and $C_4 = (\alpha_4, \beta_4)$. Then, the CLAHE of A is

$$\text{CLAHE}(A)(x,y) = \frac{\sum\limits_{i=1}^{4} |x - \alpha_i| \cdot |y - \beta_i| \cdot m_i\big(A(x,y)\big)}{\sum\limits_{i=1}^{4} |x - \alpha_i| \cdot |y - \beta_i|},$$

where m_i denotes the re-mapping function in the i-th region. Pixels on the sides or on the corners are interpolated with only two or one regions respectively. We emphasize that we consider 4 regions since we are computing the bilinear interpolation of a two-dimensional image.

Chapter 3

Review of Curvilinear Object Segmentation

Curvilinear object segmentation is a common objective accross many applications, such as medical image analysis, product defect detection or aerial image processing. In particular, vessel segmentation in retinal images, detection of spiculated lesions in mammograms or extraction of airways in CT scans provide essential information to experts to evaluate, diagnose and propose a treatment. The significance of these applications has motivated important efforts to propose curvilinear object segmentation algorithms based on different techniques. The main objective of this review is to clearly present the similarities and differences between curvilinear structures in different applications and the different techniques used to segment them more effectively. To do so, we propose a general definition of curvilinear structures that encompasses the distinct models considered in the literature. In addition, we analyse and classify the mathematical techniques used to segment the curvilinear structures found across all considered applications, studying their strengths and weaknesses. In particular, we present the most relevant benchmarks related to curvilinear object segmentation as well as the best algorithms according to several performance measures. By doing so, a wider point of view is acquired to extend the results from some fields to others, and to understand under which conditions some methodologies should be favoured over the rest of them.

The structure of the rest of this chapter is as follows. After a brief introduction motivating the study of curvilinear objects, we introduce the models of curvilinear objects according to each application in Sect. 3.2, where we also extract the common denominator of such models. In Section 3.3, the selected methods are described and classified according to the mathematical tool with which curvilinear objects are segmented. A comparison of the results provided by different methods is given in Section 3.4, along with a discussion concerning which methods can be compared and how to do such comparison. Finally, Section 3.5 concludes with a discussion, unifying the analysis scattered over the previous sections.

3.1 Introduction

Several applications extract curvilinear objects from images with different purposes. Curvilinear objects can be roughly defined as thin, long, line-like regions with different intensities than their neighbouring pixels. Such structures are found in a wide variety of situations: vessels in medical imaging, roads in satellite imagery or fingerprints from specialized acquisition devices are just a few examples of them. Since these and other applications face very similar problems, the methods that employ them tend to be based on the same techniques. However, researchers usually focus on one or two applications, disregarding the results they could potentially achieve in other fields with their own techniques. Besides not being developed in all their potential, the same mathematical tools are rediscovered and fine-grained on many occasions.

There are a great number of published methods that deal with the segmentation of line-like objects, and some highly detailed reviews of them. However, all these reviews are focused on just

one specific application. Although all of them compare and classify methods with respect to different taxonomies, there is no single study that unifies these fields to the best of our knowledge. Kirbas and Quek [92] studied and summarized techniques for vessel segmentation. More recently, a survey focused on retinal vessel segmentation was published by Fraz *et al.* [69]. Lesage *et al.* [108] presented a review of 3D vessel lumen segmentation techniques. Several methods aimed at performing airway tree segmentation in chest Computed Tomography (CT) scans were collected and compared in a study led by Lo [113]. Mena [124] published a qualitative survey of road extraction algorithms. A brief review in road pavement assessment algorithms was provided by Chambon and Moliard [46]. Other surveys on tightly related applications have also been presented, like palmprint line enhancement [97] or fingerprint classification [197]. In view of the lack of an in-depth, common analysis of these related techniques, the present survey is aimed at providing a cross-application comparison of the different strategies employed to segment curvilinear structures.

The methodology employed to study curvilinear structure segmentation has very important consequences in our study. Regarding the nature of the method, we have focused on *segmentation methods* to limit the field of work, which eases the comparison among methods and potentially empowers a more specific analysis. Segmentation methods are the ones whose input is a two- or three-dimensional image, and whose output is a binary mask indicating whether the pixel belongs to the curvilinear object or not. In contrast, enhancement methods, which may be seen as pixel-wise estimators of the belonging degree to a curvilinear object, are deliberately given a lower priority in this review. Besides, we do not include algorithms concerned with edges or contours between different objects. With regard to the studied algorithms, we find a vast amount of published works along decades when considering several applications. To increase the coverage of the survey while retaining the most relevant results, we have selected a subset of them based on, in order of relevance, the following criteria: (a) their quantitative results in the most relevant benchmarks, (b) the number of incoming cites according to Google Scholar, (c) the year of their publication, and, (d) the singularity of the application or the mathematical approach used to segment, favouring methods that segment curvilinear objects in infrequent settings. These four criteria are considered jointly instead of as four independent scores. By doing so, we have selected articles that contribute to this review due to their successful approach, their high impact on the field or their originality.

The objective of this review is to clearly present the similarities and differences between curvilinear structures in different applications and how they are segmented more effectively. By doing so, we can potentially acquire a wider point of view to extend the results from some fields to others, and to understand under which conditions some methodologies should be favoured over the rest of them.

Motivated by these two potential contributions, our goals are: (I) to select applications of curvilinear structure segmentation and study how the curvilinear objects present in them are modelled; (II) to propose a general and broad definition that encompasses the curvilinear objects that are found in all the selected fields of interest; (III) to summarize the mathematical techniques used to segment the curvilinear structures found across all considered applications; and (IV) to study the strengths and weaknesses of these techniques according to the type of curvilinear objects and to the final objective that the segmentation contributes to fulfill.

3.2   Description of Curvilinear Structures

This section establishes the foundations of a coherent study of the different mathematical techniques used to segment curvilinear structures. To do so, we should establish which objects can be considered curvilinear structures, which is accomplished by studying different applications, the algorithms that address them and their explicit and implicit models. We firstly introduce a *Model*

for Curvilinear Structures, which covers the common characteristics that share all curvilinear objects as we understand them. Besides, we present a series of applications that contain curvilinear structures, focusing on the most widely studied.

3.2.1 Model for Curvilinear Structures

Our model for curvilinear structures has been obtained as the common denominator of the explicit and implicit models considered in the selected literature.

When dealing with image segmentation, a model can be thought of as a collection of high-level features that define the object of interest, discriminating it from the rest of the image. These features must be translated into a series of constraints such that, broadly speaking, a pixel will be tagged as belonging to the object if it fulfills such constraints. Depending on their nature, such constraints will be classified, in accordance with the literature [46, 108, 103], as either geometric, when the restriction involves the spatial relations among the pixels, or photometric, in which the intensity value of pixels are taken into account.

That said, we define a *Curvilinear Structure* as a region of pixels within one image that fulfills the following geometric (**G**) and photometric (**P**) characteristics:

G1. Its pixels should be "mostly" connected.

G2. The region should be "thin" across a "long" path.

G3. The variation of width along the region should be "smooth".

G4. It should have a specific structure. That is, the overall shape of the whole segmentation seen as a binary mask. It can be a binary tree-like structure, a series of unconnected segments, a grid, etc.

G5. It should have a specific local curvature profile. This covers the behaviour of the bends that the tubular object may have. For instance, some objects may be mostly straight, others can admit soft bends and others may be highly tortuous.

G6. It should present a specific amount of bifurcations. We define a bifurcation as a three-branch joint. That is, a position of the curvilinear object where exactly three distinct branches collide.

G7. It should present a specific amount of intersections. An intersection, as we understand it, is a joint with four or more branches.

P1. Its pixels should have a "significantly different" intensity distribution compared to the distribution of its neighbouring background. In multivariate images, different channels may conceive independent information, capturing some of them more information than others.

P2. The variation of pixel intensities along the main direction should be "smooth".

P3. Its cross-section profile—the intensity values transverse to the main direction—should follow a specific distribution. In 2D images it is represented as a one-dimensional function, whereas 3D imaging has a cross section whose domain is a two-dimensional area. For example, a flat profile, also referred to as a bar-like profile, assigns a single value to each point in the cross-section.

A pixel is considered to belong to a curvilinear structure if it belongs to a region fulfilling the constraints **G1**-**G7** and **P1**-**P3**. While the contraints **G1**-**G3** remain stable among the different applications, the features **G4**-**G7** and **P1**-**P3** should be further specified to improve the model for the curvilinear structures of interest. This helps to discriminate curvilinear objects from other structures, the background texture or noise in each particular application. Tables 3.1 and 3.2 detail such features for the applications considered in this review.

We highlight that curvilinear, one-pixel width segments are intentionally included in our definition. However, this definition does not include contours or edges, since these regions do not fulfill **P1**: their intensity is similar to either one of the two regions they divide. We remark that we are not concerned with the limits between two objects, but with one tubular-shaped object placed in the foreground of a uniform or non-uniform background.

We also highlight the subjective nature of our definition, which is explicited by containing some vague words such as "mostly", "thin", etc. These vague concepts should be accurately established in each specific situation. This lack of formality is unavoidable: there are pixels that some experts may flag differently than others. Such vagueness is deliberately incorporated into our general definition.

The meaning of the aforementioned vague concepts can be illustrated with the curvilinear structures found in Fig. 3.1, 3.2, 3.3, 3.4 and 3.5. For instance, the characteristic **G1** expresses that pixels are "mostly" connected, and it remains unchanged through all applications. As we can observe across Fig. 3.2, 3.3, 3.4 and 3.5, the number of connected components is low, although not bounded by a specific number. As another example, **G5** states that the local curvature must be specific, and it is a feature that can be particularized based on the specific type of images (see Table 3.1). We observe that rural imagery contains roads composed by segments that are mostly straight or present a small local curvature (as in Fig. 3.3), whereas cracks in pavement imagery persistently suffer small, randomly distributed bends (like in Fig. 3.1(D), 3.4).

3.2.2 Applications of Curvilinear Structure Segmentation

Each application considers similar but different types of curvilinear objects, as can be observed in Fig. 3.1. For instance, vessels can be characterized as slightly tortuous tubular shapes that form a tree-like structure, whereas urban roads may be modelled as mostly straight segments with several intersections, most of them orthogonal.
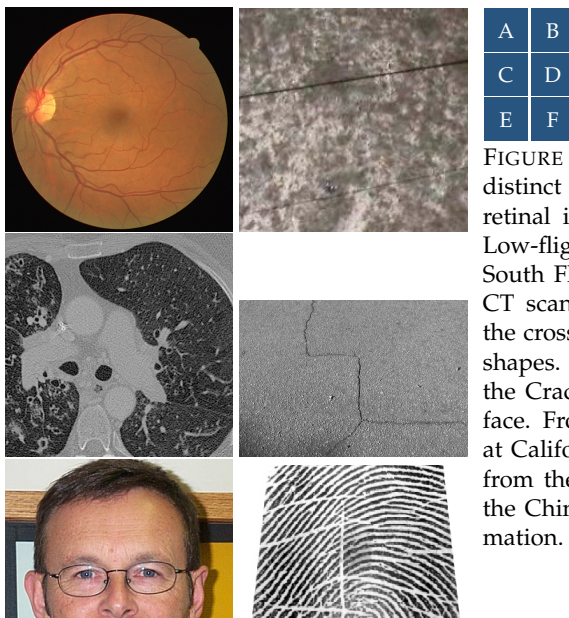
FIGURE 3.1: Samples from different datasets depicting distinct types of curvilinear objects. (A) Eye-fundus retinal image from the DRIVE database [162]. (B) Low-flight video frame containing wires from the South Florida Urban Dataset [44]. (C) Slice of a 3D CT scan from the EXACT09 database [113], where the cross-section of airways appear as black rounded shapes. (D) Pavement defect in a sample image from the CrackTree dataset [203]. (E) Wrinkles in human face. Frontal face dataset collected by Markus Weber at California Institute of Technology. (F) Fingerprint from the CASIA-FingerprintV5 dataset, collected by the Chinese Academy of Sciences' Institute of Automation.

We consider of interest the problems that (i) make use of curvilinear object segmentation according to the definition given in Section 3.2.1, and (ii) their segmentations can be contrasted against a ground truth so that the performance of different solutions are comparable. To identify the features that differentiate one application from others, we study the model of each algorithm individually. Such models, which tend to be coherent when compared to the models of other algorithms facing similar situations, are summarized in Tables 3.1 and 3.2. In the following, the applications considered in this survey are introduced.

TABLE 3.1: Geometric features **G4**-**G7**, according to the model for Curvilinear Structures from Sect. 3.2.1. We remark that the features **G1**-**G3** are stable through all applications.

	G4. Structure	G5. Curvature	G6. Intersections	G7. Bifurcations
Vessel Segmentation	Binary tree structure [11, 69] formed by piecewise linear segments [50], with the appearance of close and tangent vessels [98]. Complex loops and branching segments [176], connected tubular segments [67] or tree-like structures [4].	Small maximum curvature [69, 50].	Common and randomly distributed [176].	Common [98], highly randomly [176].
Extraction of Airways	Upside-down binary tree [144].	Non-principal curvature close to zero [144].	Not specified.	Common [113, 144].
Guide-Wire Tracking	One single object [18, 140].	Modelled as a 2nd or 3rd order curve [18, 140]. A fixed angle bounds its local curvature [35].	None [18, 140].	None [18, 140].
Road Detection	Approximable as a set of segments [177] with small width variance [202]. Segments are rarely small [202] or unconnected [165].	The angle of orientation changes is limited [165, 173, 202, 174]	Urban roads, orthogonal and frequent [165, 173]; rural roads, allowed [63].	Allowed [165], specially if their orientation changes smoothly [63, 177].
Pavement Crack Detection	Mostly longitudinal segments, transversal segments or grid-shaped structures [150, 111, 137]. They can appear unconnected [111].	Sometimes they present high curvature [203], but appear straight more often [150, 111].	Allowed [150, 111].	None [203].
Facial Wrinkles Extraction	Several continuous segments [19] with different length, thickness and orientation [54], usually segregated [19, 20].	The local curvatures is bounded by a fixed angle [19, 20].	Few [20].	None [20].

Vessel segmentation is a good example of curvilinear object segmentation. Vessel segmentation can assist in the detection of a great number of pathologies [160, 68]. It can also be used for image registration [92] and even as a biometric identifier [118]. In this type of images, both veins and arteries are curvilinear objects that should be detected and distinguished from other tissues and anatomical structures. They can be acquired with different devices. Eye-fundus images are 2D depictions that capture the retina [11, 33, 149, 160, 42, 95, 68, 183, 58, 22, 131, 146, 151, 186, 118, 125, 161, 162, 110, 199, 83, 67, 50], and can be taken with standard color photography [95, 162, 83] or with a fluorescent dye [161]. Their sizes vary from 500×500 pixels [162, 83] to few megapixels [42]. Digital Subtraction Angiography (DSA) [67], Computed Tomography Angiographies (CTA) [29, 102] and Magnetic Resonance Angiograms (MRA) [61, 145, 104, 180, 154] provide 2D or 3D representations of part of the human body. Confocal microscopy [2] and laser-scanning microscopy [176] is used for in-vivo image acquisition. These images present several types of artifacts [69, 176]. They present low contrast [50] and non stationary noise (*i.e.* that varies over the image domain) [176]. Besides, retinal images have an uneven illuminated background [11, 69, 125] and a non-flat texture [42].

Another application is the *extraction of airways* from 3D Computed Tomography (CT) scans [24, 196, 113, 187, 144, 23, 40, 64, 65, 105, 112, 172, 179, 93]. The curvilinear objects of interest are, of course, the airways. Some features of the bronquial tree structure, such as its cross-section area and

TABLE 3.2: Photometric features **P1-P3**, according to the model for Curvilinear Structures from Sect. 3.2.1.

	P1, P2. Channels and Intensity	P3. Cross-Section Profile
Vessel Segmentation	Intensity values do not change abruptly for retinal images [69], but may change in other types of images [176]. They are either brighter [125, 67] or darker [176, 50] than their background.	They present a wide range of width [11, 69, 176, 125, 67], always changing gradually along a vessel [50]. The cross-section may be modelled as a Gaussian [4, 199, 98, 50], an elliptical cross-section [176], a Difference of Offset Gaussians [125], a bar-like convolved with Gaussian [98], etc.
Extraction of Airways	The intensity is variable [144], but always brighter than the background [112].	The airway lumen has lower intensity than its walls [112, 172], which are relatively thin [172].
Guide-Wire Tracking	It is either brighter [18, 140] or darker [35] than the background.	Not specified.
Road Detection	Dark lines in SAR imaging [173], although can also appear as bright segments. In RGB imaging they can be either dark or road [177]. Higher contrast in urban areas [173, 202], whereas forest-like regions can be characterized as planar areas in Digital Elevation Models from laser imaging [63].	The statistics of the pixels are different from the statistics of the pixels situated at both sides [165].
Pavement Crack Detection	Appear as darker structures [111, 166], and tend to present a lower average intensity than the background [150, 137] and a higher standard deviation [203].	Irregular [136, 203].
Facial Wrinkles Extraction	They are darker than the background [20], and cause intensity gradients [19]. They may be extracted from the red channel [54], or from a grayscale projection [20].	They are subtle discontinuities in inhomogeneous skin texture [19].
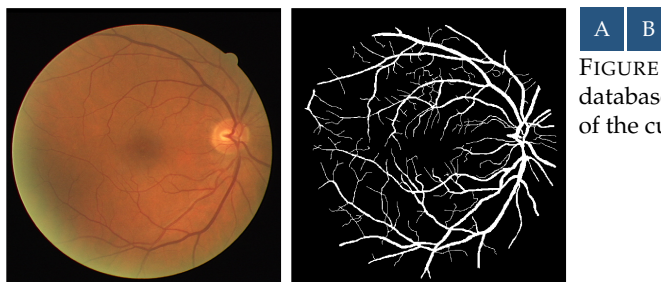
FIGURE 3.2: 6th test sample of the eye-fundus DRIVE database [162] and a mask that indicates the location of the curvilinear objects of interest, the vessels.

its wall thickness, help to detect the presence of some pathologies [130]. Their resolution is very high, around $80 \cdot 10^6$ voxels [113]. However, the lung tissue has similar intensity to airways [172], and the constrast between airways and their walls is low [144, 112]. Besides, lung diseases may magnify obstrusions [144].

During certain surgical operations, some tube-shaped devices may be inserted into tortuous cavities. *Guide-wire tracking* [35, 82] provides useful information for the practitioner to localize it within biological tissue [18, 60] or to monitor myocardial function [140]. Medical imaging provides

a series of frames from which their position can be estimated [139]. These curvilinear objects are found in 2D images [140], 2D video frames [82] or 3D images [18], and are acquired with ultrasound [18] or X-ray [35, 140] imaging devices. They are medium-sized images [82, 140] that present speckle noise [18] and low dynamic range [35, 18, 140]. They have low contrast [140], even lower when 2D images are used (since they accumulate information from the whole volume) [82]. Similarity between guide-wire and certain anatomical structures [35, 82] and distortion [18] further difficult the problem.

Road detection consists on extracting the roads, which can be modelled as curvilinear objects, from either aerial or satellite imaging [119, 188]. Regarding the models they use, they can be divided into urban [177, 165, 173, 174], rural [101, 202, 163] and forest [63, 45] regions due to their significative differences. *River extraction* [94, 57] and *bridge detection* techniques [184] face a very similar problem, and usually are complemented with each other and with road detection methods [49]. The automatic extraction of cartographic and topographic objects is a major challenge for the update of Geographic Information System (GIS) databases [124]. Besides, other structures such as ship waves appear as curvilinear objects in aerial images [89]. These applications consider only 2D [165, 202, 174] images acquired as RGB [165], RGB plus near infrared [202] or with a Synthetic Aperture Radar (SAR) [165, 173]. Forest road detection also considers the use of laser scanning data [63]. Their spatial resolution is between 7 m [173] and 0.5 m [63, 101]. They present occlusions due to vegetation or buildings [63, 177], in addition to the appearance of similar objects, like trees by the roads or shadows [177] or the presence of non-road curvilinear objects [177, 173]. In urban SAR imaging, the contrast is usually higher due to the double-bounce reflections of the buildings, although excessively high constructions may hinder the correct location of roads [173].

FIGURE 3.3: Aerial image in Mingan, Canada, and the roads found in this rural scenario. Original image licensed under the Open Government Licence by the Cadastral Information Service, Canada.

The defects in roads also appear as curvilinear objects. *Road pavement assessment* appears thus as another application that leverages our general model for curvilinear structures [136, 203, 46, 150, 111, 137, 166]. The identification and classification of these cracks [150] from specific-purpose imaging helps to automate a tedious process regarding road maintenance [166]. They are 2D images, usually being RGB images [150] or grayscale images [136, 203]. These medium-sized [203] and large [136, 150] images have a spatial resolution of approximately 1 mm [166]. They present several disturbances that diminish the quality of the images: various pavement textures [166], which are inhomogeneous [203], disadvantageous lighting conditions [150, 137] and shadows of trees or light poles [203, 166]. Oil spots may also hinder the detection of other objects [136].
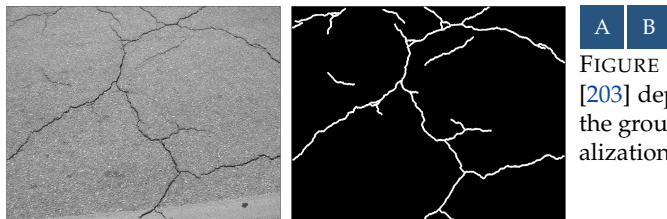
FIGURE 3.4: Sample 6240th of the CrackTree dataset [203] depicting a crack and its location. The cracks in the ground truth have been enlarged 5 pixels for visualization purposes.

Facial wrinkles detection is useful towards age estimation from 2D pictures [19, 21, 54, 20]. In them, wrinkles appear as narrow, tortuous curvilinear objects that can be considered as a soft biometrics [21] or as an age-related facial feature [54]. Images of all sizes have been considered in the literature, from small images [20] to medium-sized [19] and large ones [54]. The main challenges

are the inhomogeneuous skin texture [19, 20] and the presence of artifacts like pores, velous hairs or irregular pigmentation [19, 54].
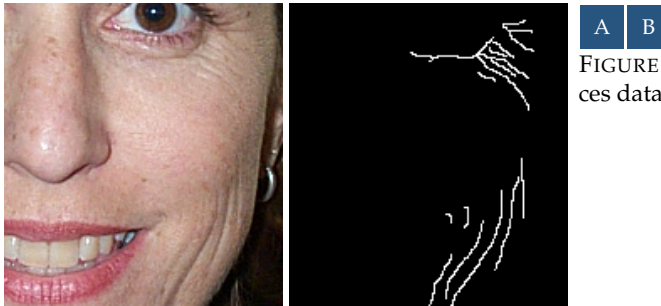
FIGURE 3.5: Detail of the 74th sample from CaltechFaces dataset, and the wrinkles found in it.

Power lines are a real threat to helicopters and small aircrafts, even in daylight [44]. *Wire detection in low-flight imagery* is an easy to implement solution that provides acceptable results [169]. Once again, these wires appear as curvilinear structures in low or medium resolution video frames [152, 43].

Neurite tracing in 2D or 3D imagery appears as another application of curvilinear structure detection. Fluorescence microscopy scans can show neurons' cell bodies and neurites: axons and dendrites that can be both characterized as curvilinear objects. The segmentation and labelling of neurites can help to study the morphology of neurons [25, 193] and axons [200, 185, 201], as well as providing neuroscientists with information to understand the molecular mechanisms in many biological activities such as neurite outgrowth and differentiation [123].

Other examples of applications that could leverage the general model for curvilinear objects are the segmentation of biological cells or filaments, such as *microglia* [195, 147], *astrocytes* [100] or *microtubules* [31], the detection of *spiculated lesions in mammograms* [205], *fingerprint or palmprint enhancement* towards matching [97, 85], the study and segmentation of *geophysical images* [141] or the detection of miscellaneous *wary objects* in natural images [107].

As can be observed in Fig. 3.1, the curvilinear objects of interest are quite diverse when considering different applications. A discussion on the model, the different applications and the features summarized in Tables 3.1 and 3.2 is found in Sect. 3.5.1.

3.3   Approaches to Curvilinear Object Segmentation

The algorithms that employ curvilinear structures can be classified as segmentation methods, which provide a binary mask to represent the segmentation; enhancement methods, that behave as curvilinear-structure enhancement filters; or centerline-extraction methods.

Each curvilinear-structure segmentation method includes—implicitly or explicitly—these three elements:

- **The Model.** The high-level constraints that describe, according to the method, what a curvilinear object is (for instance, dark and thin objects).

- **The Feature Extraction.** The numeric indicators that aim to capture one or more attributes of the model (for instance, the pixel-wise response of the image when it is convoluted with a linear filter).

- **The Algorithm.** The implementation that computes such features from the image and processes them to provide a final result (following our example, performing such convolution as a product in the Fourier domain).

This section summarizes the different mathematical tools employed by the methods that segment curvilinear objects. That is, we focus on what characterizes a method as such, which is usually

related to both its *feature extraction* and its *algorithm.* Given the wide diversity of methodologies, there is some overlapping between different categories, which are shown in Figure 3.6.
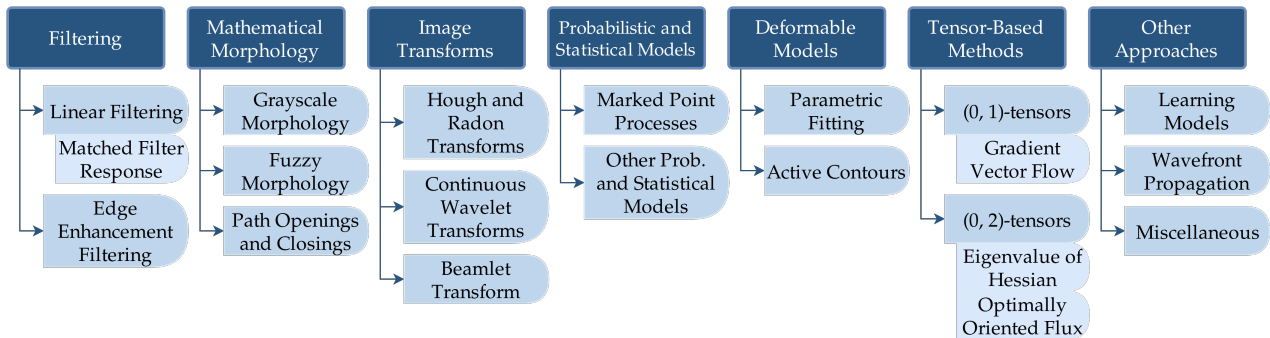
FIGURE 3.6: Categories and subcategories according to the mathematical tools that can be leveraged to segment curvilinear structures.

Besides, we also classify the segmentation algorithms according to the strategy with which they create such segmentations. This taxonomy, which is depicted in Fig. 3.7, divides the methods as:
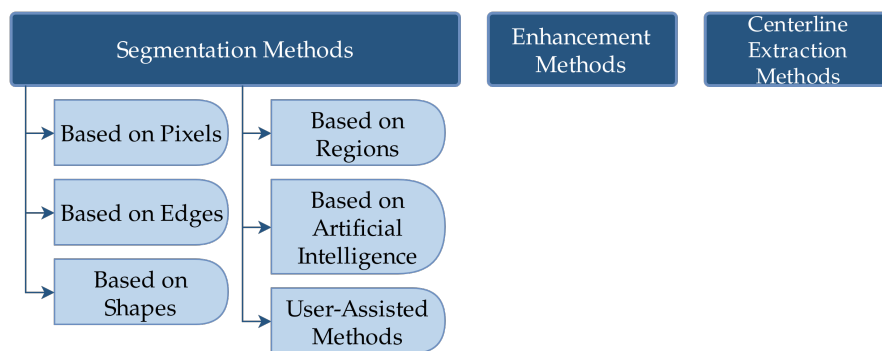
FIGURE 3.7: Categories and subcategories according to the strategy with which the segmentation can be carried out.

1. **Based on Pixels.** A method based on pixels computes a vector of features per each pixel (using local or global information), and then labels each pixel based only on its features. For instance, Huang *et al.* [85] computed several image transformations. Each pixel is then independently classified based on the value provided by such transformations in the pixel's corresponding location. This may include multiscale strategies, in which several outputs are computed depending on the size of the structure being detected and are then combined into a single output.

2. **Based on Edges.** To extract a curvilinear object, a method based on edges will first estimate the position of its borders and then fill the area within. A good example of edge-based method is the road extraction method by Zhao *et al.* [202], in which edges are computed as a first estimation. Afterwards, close pixels are assumed to belong to the same road and are, thus, linked.

3. **Based on Shapes.** This category gathers the methods that have one or more predefined templates and base their results on the similarity between the input image and these templates. Some shape-based algorithms detect a small portion of the object of interest (*e.g.* by studying

its cross section) and recover the whole structure with a tracing procedure. Besides matched-filtering methods, this approach is very well represented by the widely used "vesselness measure" [67] and similar variants [84, 155]. It consists on studying the eigenvalues of the Hessian matrix to differentiate between flat regions, blob-like structures and curvilinear structures.

4. **Based on Regions.** It includes the methods that extract a whole area at once, possibly post-processing the results afterwards. Frontwave propagation methods faithfully represent this category. For instance, the work by Van Rikxoort *et al.* [179] segments human airways starting from the trachea, which is easy to detect, and iteratively adds voxels that fulfill a series of contraints designed to discriminate the airways from the rest of structures and noise.

5. **Based on Artificial Intelligence.** It encompasses the methods in which expert data is used to infer predictions on previously unseen samples. All methods that require a learning stage fall into this category, like for instance the retinal vessel segmentation algorithm by Soares *et al.* [161].

6. **User-Assisted.** Finally, some methods require the participation of an active user, whose engagement level may vary from selecting a few seed points [40] to continuously provide feedback [172].

In the rest of this section, the methods are discussed and organized according to the first classification (see Fig. 3.6). In each subclassification, the methods are presented following their publication order.

3.3.1   Filtering

We include in this category all algorithms that convey information by using one or more filters. Since we are dealing with image processing, we understand by *filter* a local operator that, given one image as input, outputs another image of the same size. One filter, or a cascade of them, can be used to enhance desired structures found within the image. We will consider separately linear filters and the more complex edge-enhancement filters.

Linear Filtering

We will understand as linear filters those operations that can be implemented with the convolution of a specific kernel. That is, linear and shift-invariant operations. They are local operations, so they intend to detect objects with specific local properties (*i.e.* dark objects, rounded objects, edges) but they can not use global information like the position of the pixel or the image histogram. Their computational complexity is low since they are usually computed as a convolution with a given kernel. This is a fast operation when the kernel is small or separable, and can otherwise be asymptotically optimized using the Fast Fourier Transform [66].

Koller *et al.* [96] introduced a method based on a non-linear combination of linear filters, with an edge-detection approach. The local orientation is firstly estimated. Then, two linear filters are applied, each of them aimed at detecting one of the two edges of the curvilinear object. These filters are first derivatives of a Gaussian shape with an appropriate orientation. The minimum response of both filters is taken as an indicator of the degree of a pixel belonging to a curvilinear structure. This procedure is repeated at several scales, and the maximum output along them gives the final estimation.

Azzopardi *et al.* [11] presented a shape-based algorithm that leverages the difference of Gaussians, which is defined as the subtraction of the same image blurred with different Gaussian kernels.

They define a B-COSFIRE filter as the weighted harmonic mean of blurred and shifted Difference of Gaussians (DoG). The weights are computed as the ones that ouput the maximum value when processing a flat bar-shaped pattern. Multiple orientations are considered by rotating the pattern, computing the weights and applying the corresponding filter. The pixel-wise maximum response is stored across different orientations.

Matched Filter Response (MFR). MFR is a template-matching technique that leverages the fact that linear filters enhance areas similar to the kernel reflected with respect to the origin. The kernel holds, thus, the role of template. In MFR, a range of linear filters are applied, usually including rotated versions of the same filters. Afterwards, their response is merged pixel by pixel into a single result.

Chaudhuri *et al.* [50] presented a method whose filters are cropped 1-D Gaussian shapes along a direction. Further, a constant is added so their mean is zero, and they consider 12 rotations of the same kernel. Only the maximum response is retained. Hoover *et al.* [83] use this same methodology and further postprocess the results by applying an iterative algorithm to select the best threshold while facing uneven illumination, bridging gaps, preventing loops, etc. Both methods follow a shape-based strategy given the use of MFR.

Panagiotakis *et al.* [141] published an enhancement algorithm that processes geophysical images. Wavelet-based noise reduction preprocesses the original image. Their method leverages MFR: multiple filters are applied and the pixel-wise maximum response is stored. An hysteresis-based procedure is used to classify pixels as either (i) surely belonging to a curvilinear object, (ii) surely *not* belonging to a curvilinear object and (iii) uncertain pixels.

Edge Enhancement Filtering

This category includes the methods that explicitly compute the edges of the image before segmenting the curvilinear structures. Since each edge detector captures slightly different types of edges, the best one may differ depending on the situation.

Zhao *et al.* [202] segmented roads from satellite imagery with a user-assisted algorithm. Their method processes grayscale images with the Canny edge detector, which provides a binary mask. Edge pixels are linked if they are close, forming line patches. They are broken if there is a sudden change of direction, and linked together if their endpoints are close and their orientation is similar. Afterwards, a user assisted procedure uses this information and a road-likeness estimation provided by a commercial software to create the road map.

Candamo and Goldgof [43] segmented wires from low-altitude aircraft flight. They computed edges as a combination between the Canny edge detector and a series of morphological operations. Since wires are assumed to be straight, one-pixel width lines, this edge map is postprocessed with a line-fitting procedure to extract the wires. The same authors extended their method by considering the Hough transform as a postprocessing algorithm [44]. These two algorithms employ an edge-based strategy given the importance of the edge detector. The line-fitting procedure and the Hough transform are introduced in detail later in this section.

Lemaitre *et al.* [107] presented a general curvilinear region detector to detect wiry objects in multiple types of natural 2D images. At multiple scales, the edges are detected and linked by searching for pairs of edges taking into account their geometric properties. A series of postprocessing operations remove potential false positives and duplicates on different scales. Bifurcations and crossings of curvilinear objects are explicitly considered. This postprocessing ensures that the selected regions have regular shapes.

3.3.2   Mathematical Morphology

The operators of mathematical morphology are simple, local and non-linear operations that, when properly combined, provide complex and elaborated results. We will consider as morphological those methods that integrate these basic operations towards segmentating the objects of interest. They are often based on two operators: the dilation $\mathcal{D}$, that enlarges bright areas, and the erosion $\mathcal{E}$, that diminishes them. A wide variety of operators can be defined by composing them, either sequentially or iteratively (*e.g.* until no changes are observed). Although the foundations of mathematical morphology consist on simple operators, they provide complex results when properly combined. Due to their local behaviour, they are suited to detect specific objects (like skeletons, convex hulls, etc.) but they are limited with regard to the global information they can use. When compared to linear filters, they have more expressive power but higher computational requirements.

Grayscale Mathematical Morphology

It is a natural extension of binary mathematical morphology, originally developed for two-dimensional black-and-white images, to deal with single-channel grayscale images. One of its most common extensions to deal with grayscale images is the threshold approach [129, Sect. 3.1], in which flat structuring elements are used to define the grayscale erosion and the grayscale dilation.

Zana and Klein [199] presented a pixel-based method using grayscale mathematical morphology. Their algorithm enhances pixels belonging to linear segments by computing the sum of 12 directional top-hat operations, defined as the original image minus the dilation of its erosion. The direction is encoded into the structuring element. They further enhance the curvilinear structures by studying their curvature. Noise is removed with both a preprocessing operation and a postprocessing filtering, both of them also based on morphological operations.

Mendonça and Campilho [125] introduced a shape-based algorithm that consists on (i) ridge detection, (ii) morphological curvilinear-object enhancement and (iii) the reconstruction of curvilinear objects. For (i), they impose the difference of offset Gaussian cross-section with a matched-filtering like procedure. They search for maximums at multiple scales, obtaining the centerlines. For (ii), they use a modified version of the top-hat transform. They firstly compute the closing (erosion of dilation) of the image with a circular structuring element of radii equal to 1 pixel. From this result, they compute the opening (dilation of erosion), with circular structuring elements of different sizes to handle vessels at multiple scales. The modified top-hat is the subtraction of the original image minus the opening of the closing, or zero if this provides a negative value. The top-hat transform at each scale is processed with an hysteresis operation. In (iii), the results of (i) and (ii) are merged by iteratively combining the centerlines with the vessels enhanced at multiple scales.

Fetita *et al.* [64] extracted airways from CT scans with a region-based morphological approach. Their method uses the so-called flood size-drain leveling morphological operator. It generates a brighter image and then applies a morphological reconstruction by erosion with the original one. This procedure finds the curvilinear objects from their walls by filling the "holes" of the latter. The segmentation is postprocessed, retaining only the segments connected to the trachea.

Fraz *et al.* [68] introduced a shape-based method in which centerlines are detected with a First order Derivative of Gaussian (FoDoG) filter applied in four directions. Vessels are then reconstructed by means of the top-hat transform and a region-growing reconstruction. The summation of morphological top-hats with directional structuring elements (specifically, 8 rotations of a bar-shaped mask) is used to create the map of blood vessels. Only the information of its two highest bits is used, since they contain almost all the visual information. Finally, the pixel-wise segmentation is provided by a binary reconstruction procedure. The centerline pixels are the seeds whereas the growing mask is composed of the pixels such that at least one of its two highest bits of the map of blood vessels is equal to 1.

Roychowdhury *et al.* [149] presented an iterative method based on region growing. They obtain a first estimation by performing a top-hat transform (the opening subtracted from the original image) of the inverted green channel, obtaining a rough estimation by thresholding it. The iteration consists on selecting as seed points both (i) the whole previous iteration and (ii) some new pixels from the top-hat transform (according to an iteration-dependent threshold). These seed points are set to 1, and then a hysteresis is performed with $t_{\text{high}} = 1$ and an iteration-dependent t_{low}. The stopping criterion tries to maximize the accuracy of the final estimation by predicting the number of false positives, false negatives, true positives and true negatives from the previous and actual iterations.

Fuzzy Mathematical Morphology

It is a paradigm that generalizes the binary mathematical morphology while dealing with the uncertainty found in the acquisition or postprocessing of images [55]. Assuming that grayscale values range in $[0, 1]$, it uses fuzzy logical connectives along with multivalued structuring elements to perform the basic morphological operations. Specifically, the fuzzy erosion and the fuzzy dilation are defined considering grayscale structuring elements, fuzzy conjunctions and fuzzy implication functions [55]. For more details, see Sect. 2.4.

Bibiloni *et al.* [33] presented a method in which, following the work in [199], the curvilinear structures are enhanced with a top-hat transform. In their method, they use a Gaussian-shaped structuring element, the minimum operator as a fuzzy conjunction and its residual implication. They use a local histogram equalization for contrast enhancement and a hysteresis procedure to prioritize connected components. Due to the behaviour of the hysteresis, it becomes a region-based algorithm.

Path Openings and Closings

They are morphological operators that were specially developed to detect and extract curvilinear structures. Originally introduced by Heijmans *et al.* [81], path openings and closings extract the pixels that are connected with at least L other pixels constrained to a directional adjacency matrix (*e.g.*, they restrict the possible connected pixels to the upper and right pixels in their 8-neighbourhood). Besides, they have been adapted to detect incomplete paths and to be used in high-dimensional settings [53, 168]. It is classified as a morphological operator since computations are made locally and the results are indeed morphological openings and closings. They only allow a small deviation with respect to a straight line. As an extreme example, a spiral would be recognized as a large number of tiny, nearly straight segments. Several algorithms have been developed to compute them with an acceptable efficiency. However, several adjacency matrices should be used to detect lines with different orientations.

Valero *et al.* [177] presented a road segmentation method based on morphological profiles. They are pixel-wise vectors that represent different results obtained when sweeping a parameter. In their case, the morphological profile represents the output of grayscale path openings and path closings with different minimum lengths. It is used to extract an indicator of the length of the path that crosses each pixel, which is thresholded to obtain the road pixels, becoming thus a pixel-wise segmentation method. Such indicator is estimated as the minimum length such that the grayscale level provided by the path opening or the path closing is below the median gray level of road pixels in the image.

Sigurðsson *et al.* [160] extracted retinal vessels with a region-based algorithm. The preprocessing consists on a Contrast Limited Adaptive Histogram Equalization (CLAHE), a 3×3 Gaussian smoothing and a small opening intended to remove small and isolated points. Two features are

extracted: the black top-hat, to extract locally dark regions, and the Difference of Gaussians, to enhance edges. A third feature is provided by a path opening algorithm, intended to recover oriented and connected curvilinear objects. Finally, these features are interpreted as fuzzy sets and fusioned into a single indicator based on the multiplicative class.

3.3.3   Image Transforms

Other approaches decompose the original image into a basis, yielding one coefficient per element of the basis that, broadly speaking, indicates *how much* the basis element is found *within* the original image. Although mathematically speaking they should always admit an inverse, we also include some transformations that do not fulfill this property but just describe the image as a series of coefficients. These coefficients can be seen as another representation of the original image that yields in an abstract space, the transformed space. This representation may expose useful properties that were hard to extract otherwise. Both their computational requirements and their expresiveness depend on the actual transformation used, although both tend to be high.

Hough and Radon Transforms

The Radon transform computes its coefficients as the line integrals along each straight line, and it is applied over two-dimensional functions with a continuous domain. The original Hough transform, which handles discretized images, is tightly related to the Radon transform. It provides the summations along each straight, one-pixel width line. The latter can be seen as a discretization of the former, although it was independently designed to detect straight lines in black and white images [70]. In both cases, the considered lines range over all the possible locations and orientations, and are not bounded except for the image limits. The original Hough transform is efficiently computable and a good estimator to detect straight lines, but it is not suited to detect curved structures or short segments.

Palti *et al.* [140] use the Hough Transform to fit a second-degree curve in a sliding window, providing a shape-based method to detect a guide-wire. After preprocessing the image, the Hough transform is applied to find the curve of the form $y = ax^2 + bx + c$ that yields the greatest sum along its pixels. This curve fitting mechanism is applied in a 71×71-pixel window, and its result is used to compute the solution in the next window, with which it overlaps partially.

Huang *et al.* [85] extracted the curvilinear wrinkles in human hands to calculate the similarity between two palmprints with a pixel-based method. They introduce a version of the Radon transform that they call the Modified-Finite Radon Transform (MFRAT) to avoid the periodicity assumption of the Finite Radon Transform (FRAT). It considers bounded segments in a given grid instead of lines. It provides a local best-line estimator, including its orientation and the energy of each segment (*i.e.* a measure of the certainty with which the estimator is indeed a true positive). Besides, a threshold and other postprocessing operations are applied.

Krylov and Nelson [99] published a shape-based algorithm to segment curvilinear structures. They divide the image into an overlapping grid of square regions and use the finite Radon transform to find segments in them. They define a pixel-wise energy term and remove weak candidates by applying a threshold.

Continuous Wavelet Transforms

Continuous Wavelet Transforms (CWT) constitute a family of transforms that provide a different insight of the information contained within an image. These transforms are carried out with

wavelets: functions that begin and end at zero—leading to a local behaviour—with a wave-like shape that provides information about the "local frequency" of the images. Intuitively speaking, information that explains how the image evolves in a certain neighbourhood. Although "local frequency" may seem contradictory in itself, these transformations employ a sliding window, big enough to contain frequency information but small enough to separate different parts of the image. The Morlet wavelet is, essentially, a complex exponential windowed with a Gaussian function [10]. A more elementary version of it, often referred to as the Modified Morlet Wavelet or Gabor Wavelet, is not invertible neither normalized [52], but is conceptually simpler. Other families are generated with distinct mother wavelets, which can be seen as the prototype used to define the elements of the basis.

Subirats *et al.* [166] detected cracks in pavement using the Continuous Wavelet Transform. Specifically, they use the 2D CWT rotated $0°$ and $90°$ at different scales with the Mexican hat function as mother wavelet. At each scale, this setting provides two features with which they extract a maxima location map. Afterwards, these maxima maps are merged into a single-scale map from the lowest resolution to the greatest one. Finally, the binary output is computed by thresholding, providing thus a pixel-based method.

Batool and Chellappa [19] presented an algorithm that enhances curvilinear structures with Gabor filters to detect facial wrinkles. The output of each Gabor filter is normalized across the pixels of the image. Then, the enhanced image is created by assigning to each pixel the maximum response of the normalized Gabor filters. Afterwards, they use this enhanced image to trace wrinkles in them with a region-growing approach, employing thus a region-based segmentation strategy. A further refinement reduces false positives. Cula *et al.* [54] also used Gabor filters to detect wrinkles. In their case, the local orientation is estimated in a local window. Only one Gabor filter is applied at each location, oriented according to the local orientation and manually tuned.

Beamlet Transform

Another technique based on wavelets is the beamlet transform, introduced by Donoho and Huo [59]. The beamlets are the elements of the basis of this non-invertible transform. They are a subset of the segments contained within the image domain (the so-called beamlet dictionary). Each coefficient is then computed as the integral over the corresponding segment. Instead of considering all possible segments between all pairs of vertices (which would include a huge amount of segments, adding up to $O(n^4)$ for $n \times n$-pixel images), in each scale only the segments that connect two close vertices within a subgrid are considered (which yields $O(n^2)$ segments).

Berlemont and Olivo-Marin [30] used a region-based strategy with beamlets to segment curvilinear structures in noisy images. To compute the beamlets, they use a method based on the Fast Fourier Transform after the use of steerable filters on the original images. They normalize the coefficients and select the beamlets that provide more information. Afterwards, the significant beamlets are linked to each other to obtain the final set of curvilinear objects.

3.3.4 Probabilistic and Statistical Models

This category embraces the methods that base the detection of curvilinear objects on the use of probability theory, statistical distributions or statistical tests. Although most methods use probabilistic tools (*e.g.* when estimating the variance of an indicator), we only consider within this category the ones whose nature is, almost entirely, probabilistic or statistical. Both their expressive power and

computational requirements depend greatly on the specific technique. However, given that they are theoretically well founded, they tend to be very expressive.

Marked Point Processes

A Spatial Point Process is a probabilistic model of the distribution of a (usually finite) number of points in $S \subseteq \mathbb{R}^d$. A Marked Point Process is a spatial point process in which each point has some *metadata* associated with it, so instead of considering realizations as $\mathbf{x} \subset S$, they will have the form $\{(x_1, m_1), \ldots, (x_n, m_n)\} = (\mathbf{x}, \mathbf{m}) \subset S \times M$. Typically, each point (x_i, m_i) corresponds to a segment, in which x_i is its center (so $S \subset \mathbb{R}^2$, that corresponds to the scene), and $m_i = (w_i, l_i, \theta_i)$ are its width, length and orientation, respectively.

We assume that the real scene is digitized on a finite pixel lattice, the digital image. We consider points as being segments in the scene (being the coordinates of the center, the length, the width and the orientation real numbers) and decide which set of segments represents best the scene by considering the only information we have about it: the discretized image. Marked Point Process techniques pick the best combination (according to a probability measure) of marked points—or segments—that represent the original scene. To do so, a solution is provided by minimizing an energy function, which is composed of two terms: the data energy, which depends on the image, and the interaction energy, which only depends on the marked points. This last term can be seen as the prior probability model since it introduces the purely geometric constraints (*e.g.*, specifying how marked points may intersect with each other).

In [165], Stoica *et al.* use a Gibbs point process to segment roads from remotely sensed imagery. That is, one in which the energy computed from the image is taken into account with a Gibbs measure. As data energy term, they consider a segment to be part of the network of curvilinear structures if the average intensity of the pixel's segments are different from those situated at both sides. The interaction energy term penalizes short, unconnected and overlapping segments. The region-based solution formed by the selected set of segments is computed with the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm. Lacoste *et al.* [102, 101] extended this work by considering gradual coefficients to describe the satisfiability of a condition and an improved data energy term that considers the homogeneity of regions. Ferraz *et al.* [63] also used the same statistical model to segment roads in forests.

Batool and Chellappa [20] used a Gibbs point process to extract wrinkles from human faces with the same region-based approach. The "data energy" they use is the output of a Laplacian of Gaussian filter, and their "interaction energy" rejects congested areas. They also use the RJMCMC algorithm.

Other Probabilistic and Statistical Models

Draper *et al.* [60] introduced a pixel-based method that localizes the needle in breast biopsies. To face the speckle noise and enhance the needle, their method estimates the variance in each pixel's neighbourhood, which describes the nature of the whole method. Since the needle is much brighter than the background, its pixels will present high variance. This variance is computed in a squared window in which all pixels have the same influence. Some postprocessing operations, including a binary closing to smooth the edges of the needle detected, provide the final segmentation.

Narasimha *et al.* [131] used statistical hypothesis tests to discriminate vessels in retinal eye-fundus images. Their region-based method labels pixels as being part of a vessel according to a hypothesis test, in which H_0 indicates the absence of a vessel through the pixel, and H_1 otherwise. For doing so, they consider two features of the model of vessels: (i) they should be locally linear with parallel edges along a particular orientation and (ii) their cross-section profile should be specific. They consider a variety of cross-section profiles to detect vessels with central reflex—a

disturbance by which a bright region appears in the centerline of the vessel. The Dual-Gaussian model [110] is the profile that yields the best results. This model was rediscovered by Xiao *et al.* [191] under the name of Bi-Gaussian function. It generalizes the Gaussian kernel and obtains a behaviour more similar to that of a rectangle kernel. Besides handling the central reflex, it also provides a good trade-off between noise suppression and separation of close structures. The cross section profile is defined as a piece-wise continous function composed of two Gaussians, one for points close to the center and another Gaussian shape, scalated to preserve continuity, describing the rest of points.

3.3.5 Deformable Models

In this category we include all methods in which a model is adapted to fit the curvilinear structure represented in the input image. Depending on the approach with which the model is adapted to the image, they are further subdivided into *Parametric Fitting* techniques and *Active Contours*.

Parametric Fitting

It includes the techniques in which a usually small number of parameters control a predefined model. Whereas some aspects of the model may be modified (*e.g.* size, eccentricity, etc.), others remain fixed (for instance, its tubular appearance). Often, a local model is used to detect if a small area of the image belongs to the object of interest. The whole segmentation is, thus, given as the union of all chunks fulfilling the model, which are usually found with a tracing algorithm.

Abdul-Karim *et al.* [2] considered a model that generalizes 3D cylinders: those regions in which the 2D cross-section profile present a specific behaviour in its four directions $(-x, +x, -y, +y)$. Local maximum intensity voxels provide seed candidates. They are the starting points for tracing, which continues until the model is violated, following thus a shape-based strategy. Among other features, the tracing procedure takes into account the distance between consecutive centerlines and their angle.

Barva *et al.* [18] model a curvilinear object as a three-dimensional cubic curve and use the randomized RANSAC algorithm [51] to find the best match. Repeatedly, four points are randomly selected and interpolated with a cubic curve. This curve is used to estimate inliers and outliers, and a curvature measure is taken into account to avoid excessive oscillations. The quality of each solution is estimated, and the whole procedure ends when the probability of finding a better model is low. Afterwards, they refine the method by estimating the endpoints based on statistic-based parameters. It can be classified as a region-based algorithm.

Tyrrell *et al.* [176] segmented curvilinear structures with an algorithm based on shapes. The deformable model they use to extract such shapes is the Cylindroidal Superellipsoid: an explicit, low-order parametrization that provides the centerline and border location of a curvilinear segment. A generalized likelihood ratio test is used to distinguish superellipsoids that represent curvilinear structures from the ones containing noise and artifacts. To combine local segmentations, a tracing algorithm is executed in a greedy way while refining the cylindroid parameters.

Wang *et al.* [186] presented a shape based algorithm with an unusual but powerful multiscale approach. Firstly, the image is divided as a series of nested grids, which provides overlapping square regions. In each region they estimate the parameters of their model, $f(x) = G(x) + B(x) + n(x)$, where each segment of the curvilinear object $G(x)$ is modelled as a 2-D bivariate Gaussian

kernel, the background $B(x)$ is thought of as a piece-wise linear region and $n(x)$ is additive Gaussian noise. A subset of disjoint regions are selected with a bottom-up approach from their over-represented grid, prioritizing the scale at which the model best explains the subimage. The curvilinear binary tree is segmented with a Markov Chain stochastic process that maximizes a posterior probability.

Padoy and Hager [139] used a spline derived from a set of points to create a region-based algorithm in a sequence of stereoscopic video frames. More specifically, they use a non-uniform rational basis spline (NURBS). They handle it as an energy minimization problem, with a term to enforce the curvilinear shape (distance to closest ridge) and the texture (distance to closest pixel with specific color), and a second term enforcing the in-extensibility (constraining local changes of arc-length in subsequent frames). This tracking procedure is a region-based algorithm.

Active Contours

This subcategory includes all those methods in which the solution is provided by means of an elastic route obtained by iteratively minimizing a certain function. This function is thought of as the *energy* of the solution, and it is typically affected by different terms (*e.g.* the amount of connected components of the estimation, how they are related with the gray values, etc.). Snakes [90] constitute a well-known example of active contours. They are active contours that are pushed by several forces, both internal and external. The internal force only depends on the snake itself, and usually enforces smooth contraints: they include tensile forces (to uniformly distribute the nodes) and flexural forces (to control the snake's bending). External forces are those that couple the snake to the image.

Related approaches include active contours based on other vector fields, like the Gradient Vector Flow (GVF), which is classified as $(0, 1)$-tensors later in this section.

McInerney and Terzopoulos [121] introduced Topology Adaptive Snakes or simply T-snakes. A T-snake is a set of nodes which, connected in series, form a closed 2D contour. Unlike traditional snakes, a Topology Adaptive Snakes dynamically changes its nodes and interconnecting elements. The image is decomposed into a grid of cells (the so-called ACID grid). The evolution of the snake is divided into two phases, which are alternated. The first one corresponds to the Lagrangian dynamics phase, and during the second one, the T-snake is reparametrized by computing the intersection between the T-snake and the ACID grid. This last phase, the topological one, provides the adaptability needed to segment complex structures. Besides, to avoid loops, the model is not allowed to go backwards.

Al-Diri *et al.* [58] segmented vessels in retinal images with a related active contour: the ribbon of twins. Each ribbon of twins contains four active contours, and tries to capture both edges of a vessel. The two inner contours are linked with a force to maintain vessel width consistently, whereas each of the outer contours move towards their respective inner one. Since this model only works well with segments, they have to be joined afterwards with some postprocessing procedures. Besides, the initialization of each ribbon of twins needs some clues about its approximate location, needing thus a rough estimate of vessel centerlines as a preprocessing step. Given the nature of this active contour, this constitutes an edge-based method.

3.3.6 Tensor-based Techniques

Tensors are a generalization of scalars ($s \in \mathbb{R}$), vectors ($v \in \mathbb{V}$) and linear maps of the form $\ell : \mathbb{V} \to \mathbb{R}$, also called covectors. In our case, we consider the vector space $\mathbb{V} = \mathbb{R}^d$ when dealing with

d-dimensional images. More formally, let $\mathbb{V}$ be a vector space and let $\mathbb{V}^*$ be the set of covectors of $\mathbb{V}$. Then, we say that a (n, m)-tensor T_m^n (or *n*-contravariant, *m*-covariant tensor) is a multilinear application $T_m^n : (\mathbb{V}^*)^n \times \mathbb{V}^m \to \mathbb{R}$, as defined in [3]. That is, such tensor provides a scalar whenever it is evaluated with *n* covectors and *m* vectors as arguments.

Mathematically, we can describe an image as a scalar-valued function $f : M \to \mathbb{R}$, where *M* is a 2D or 3D discretized grid. In addition to the previous concepts, we define a (n, m)-tensor field on *M* as an assignment of one (n, m)-tensor to each point in *M* [3].

The generalization that tensors represent helps us relating different concepts. For instance, a vector field can be seen as a $(1, 0)$-tensor field, and the image itself is a $(0, 0)$-tensor field. Given that first and higher-order derivatives are linear operations, the notion of tensor is appropriate to describe them. For example, the first derivative is a $(0, 1)$-tensor field since, for each direction $v \in \mathbb{V}$, it provides a scalar representing the sensitivity to change in such direction. The second derivative can be similarly described as a $(0, 2)$-tensor field.

(0, 1)-Tensors

They are methods focused on a specific $(0, 1)$-tensor field. The gradient, which is an example of $(0, 1)$-tensor field, can be approximated with the help of convolutions. Let $G(\mathbf{x})$ be a smoothing Gaussian shape and *A* an image. Then, the gradient can be approximated as

$$\frac{\partial}{\partial x_i}\Big[A(\mathbf{x})\Big] \approx \frac{\partial}{\partial x_i}\Big[A(\mathbf{x}) \ast G(\mathbf{x})\Big] = A(\mathbf{x}) \ast \frac{\partial}{\partial x_i}\Big[G(\mathbf{x})\Big],$$

where $\ast$ denotes the convolution operation.

Since the derivatives of the Gaussian shape *G* can be precomputed, these operations are fast and do not introduce further numeric errors. Besides, the variance of the Gaussian shape can be incremented to remove the effects of noise, since more pixels contribute to compute the same derivative; or diminished, so that sharp and close areas are easily detected. This is a tuneable parameter to procure a good trade-off between exactitude and robustness.

Vasilevskiy and Siddiqi [180] defined a flow using the gradient of the image as a vector field, and used it to extract vessels in different angiography image types. The segmentation is circumvented by a closed curve. It evolves so that the inward flux, along with some regularization parameters, is maximized. Since the segmentation is then set to be the inner part of the curve, it can be considered an edge-detection based algorithm.

Sanders-Reed *et al.* [152] detected wires in low-flight imagery with a pixel-based algorithm developed in Boeing-SVS, Inc. After some noise-reduction preprocessing, the pixel-wise gradient is estimated with two 5×5 convolution kernels. Per each pixel, the method adds the gradient of its neighbours weighted by their corresponding mask value $K_n = \pm 1$. Depending on the resultant vector, the pixel is classified as uniform gradient, line or noise area. To fill holes, 16 orientations of a 41×41-pixel bar-like kernel are considered. If enough pixels are found within some rotation of the bar-like kernel, the central point is also set to 1. Besides, temporal and other spatial filters are applied to further refine the results.

Pu *et al.* [144] segmented the human airway tree with a shape-based segmentation algorithm. Firstly, the lung volume is estimated and kept as mask (with a histogram-dependant thresholding). The nature of the method is given by the curvature of 3D airways, computed by means of the finite-difference analysis of normal vectors, which yields reasonable accuracy and low computational complexity. The curvature is used to discriminate objects that are not cylindrical. A postprocessing step removes false positives and false negatives by reconsidering the labels of isolated regions with respect to the whole segmented tree.

Zou *et al.* [203] segmented road cracks with a tensor voting algorithm. Firstly, they apply a

texture-preserving illuminance compensation to compensate any shadows, a geodesic shadow-removal algorithm (GSR). It consists on a morphological closing followed by a 2D Gaussian smoothing, a generation of contour-lines with N thresholds, and an illumination compensation step. Secondly, they perform detection with a local intensity-difference measure and tensor voting. The former is achieved by comparing the intensity of one pixel with its 8 neihbourhood and extracting those values lower than a threshold that depends on the global histogram. Tensor voting finds principal directions according to surrounding pixels, considers an oriented cone at each point and enhances curvilinear objects that may appear within the cone, being thus a shape-based approach.

Ferraz *et al.* [63] detected roads in forests using Digital Elevation Models (DEM). Their characterization is based on the slope and the aggregated multi-scale slope, as well as in the homogeneity of the regions. Using random forests and these three pixel-wise features, their method estimates whether a pixel belongs to a road segment. The final estimation is computed by means of the global statistical model from Stoica *et al.* [165], so the method provides region-based segmentations.

Gradient Vector Flow (GVF). Introduced by Xu and Prince [194], the GVF is a vector field obtained by minimizing an energy functional over the whole image. It is a vector field similar to the gradient but aimed at overcoming some of its weaknesses: the GVF has a wide range of capture (*i.e.* points far from edges are still attracted to them) while is still influenced by pronounced concavities (*i.e.* points with close edges are still slightly affected by further edges). It can be considered an edge-preserving diffusion of the gradient vector. Besides, it requires an appropriate initial vector field where the vectors point towards the centerlines of the structures of interest and whose magnitude estimates the edge-likeliness.

Bauer and Bischof [22] use the GVF in their pixel-wise method. They compute the derivatives of the GVF and estimate if a pixel belongs to a curvilinear structure using its eigenvalues, in a similar way to [67]. A simple thresholding provides the centerlines of the curvilinear structures. Bauer *et al.* [23] extended their work to reconstruct the whole curvilinear structure from the centerlines. It consists on a region-growing procedure that uses the fact that the gradient magnitude increases until the boundary of the object is reached.

(0, 2)-Tensors

Some methods use more information than just a vector per pixel. This subcategory includes the techniques that use $(0, 2)$-tensors, which can be represented as a matrix per each pixel. Such amount of information increases the computational requirements but potentially provides better quality segmentations.

Hessian matrix and analysis of its eigenvalues.   The Hessian, which may be estimated by convolving the image with the derivative of a kernel, conveys important information about a pixel's neighbourhood.

Steger [163] models lines as 2D curves with a specific profile. They detect the orientation of a line from the eigendecomposition of the Hessian, and estimate its strength as the absolute value of the second directional derivative across the detected orientation. They later link points to create lines, based on the orientation of the line, its strength and the sub-pixel location of the potential line. This is a greedy procedure that employs a shape-based strategy.

Frangi *et al.* [67] popularized the use of the Hessian matrix and the analysis of its eigenvalues. Such eigenvalues describe the principal directions in which the second-order structure of the image can be decomposed. With 3D images, a tubular segment is characterized if and only if one eigenvalue is small in absolute value and the other two are large in absolute value. They introduce a measure, the so-called *vesselness measure*, which is based on the eigenvalues and captures this behaviour, which is a pixel-wise estimator of the belonging degree to the curvilinear structure.

Since they do not provide a binary segmentation, their method can not be classified according to its segmentation strategy. It is a curvilinear object enhancement filter, and has been used by several authors [100, 42, 193, 123]. The work is extended by Sato *et al.* [155], who provided a tightly related curvilinear measure, and by Huang *et al.* [84], who used a single-scale filter to compute the Hessian matrix, to address the problem of separating close tubular structures in noisy images. Xu *et al.* [196] used the vesselness measure along with other information to improve the quality of their segmentation.

Staal *et al.* [162] presented a shape-based method that locates the ridges and then reconstruct the whole curvilinear structure. The ridges are situated where the sign of the gradient changes along the direction of largest curvature, given by the eigenvector with largest eigenvalue, in absolute value, of the Hessian matrix. They form ridges with a growing algorithm based on a scalar field that takes ± 1 for ridges of local minima/maxima, and 0 elsewhere. Then, several pixel-wise features basic photometric and geometric features are computed: R, G, and B values; estimation of the region of the pixel being a vessel, etc. The binary output is obtained as the output of a kNN classifier.

Zhang *et al.* [200] firstly detect centerlines with a gradient vector tracking procedure, using their generalization of GVF. Afterwards, to refine the results, they use the eigenvalues of the Hessian matrix to distinguish different but close curvilinear objects. To do so, they use the assumption that different objects will have different shapes, which are distinguished by the aforementioned eigenvalues. Since this procedure provides the centerlines of the objects of interest, the Fast Marching Algorithm [159] completes their region-based segmentation.

Optimally Oriented Flux (OOF). Introduced by Law and Chung [104], OOF is a (0,2)-tensor that eases the computation of the inward flux in the direction in which it is minimum. Conceptually, they define a sphere with a certain radii and study the flow across the sphere in each direction. Computationally, however, it is reduced to several convolutions, which can be performed efficiently. It intends to separate adjacent structures, which is a weakness present in the gradient or Hessian matrix computed with the derivatives of Gaussian shape. A multiscale analysis is possible by varying the radii of the sphere.

Law and Chung [104], besides introducing the OOF, introduced a method to segment curvilinear structures. They compute the eigenvalues of the 3×3 OOF matrix that provides such flux. Analysis of these values provides the pixel-based classification.

Benmansour and Cohen [29] segmented different types of tubular structures using the OOF with a region-based strategy. They define a pixel-wise anisotropic metric based on the eigenvalues of the OOF matrix, in a similar way to Frangi *et al.* [67]. Besides the original 2D or 3D domain, they add an extra dimension that represents the radius of a line. Therefore, a curve in their extended domain represents a curvilinear object in the original domain. The segmentation is provided as a minimal path in the extended space with the anisotropic metric.

3.3.7 Other Approaches

In this last category, we group those methods that do not fall into any of the previous categories. This includes methods based on artificial intelligence approaches, algorithms based on growing a certain region with a wavefront propagation strategy, and finally a series of diverse methods.

Learning Models

This last category encompasses the learning methods that base the segmentation on a pixel-wise automatic classification. They often consider a massive number of features that, although being excessively large for an expert system, is suited to be used with an automatic learning approach. The computational complexity depends on both the feature calculation—where the expressive power lies—and the classifier itself. Therefore, each individual method achieves a different trade-off between its ability to correctly detect curvilinear objects and its computation speed.

Soares *et al.* [161] introduced a method that leverages features based on oriented multiscale Morlet wavelets and a Bayesian classifier called Gaussian mixture. In a pixel wise fashion, they consider 18 orientations, two scales and one eccentricity. With each combination, they apply the Gabor wavelet, obtaining a different feature, which is normalized afterwards. Each pixel is classified with a Gaussian Mixture Model, in which a linear combination of Gaussian functions describes each class-conditional probability density.

Ricci and Perfetti [146] used Support Vector Machines (SVM) to segment vessels in retinal images. The features they consider are (i) a basic line detector—a simple, non-linear operation along a line-shaped structuring element at 12 different orientations—; (ii) same as (i) but with a cross-shaped structuring element; and (iii) the gray level intensity of each pixel. Features are normalized and used as input to a linear SVM. In addition, the cost of false negatives with respect to the cost of false positives is compensated given the asymmetry of classes.

Salem *et al.* [151] published a human-assisted method that segments retinal blood vessels using a clustering algorithm. The features considered are (i) the green channel intensity, (ii) the local maxima of the gradient magnitude at multiple scales and (iii) the local maxima of the largest eigenvalue at multiple scales. The clustering algorithm employed is the Radius based Clustering Algorithm (RACAL). Essentially, two samples are assigned to the same cluster as long as their distance, in the feature domain, is lower than a given threshold. Their method constitutes an interesting human-assisted strategy: with the clustering technique, they select only a reduced amount of significative samples, reducing thus the amount of human interaction required.

Lo *et al.* [112] introduced an algorithm that uses the k-Nearest Neighbours learning procedure. They consider several local features, including the spatial first and second order derivatives, the eigenvalues of the Hessian matrix and several indicators computed from them. The rate of neighbours classified as vessel, along with the orientation derived from the Hessian matrix, are taken into account to segment the curvilinear structure in a region growing fashion.

Honnorat *et al.* [82] introduced an algorithm to extract a guide-wire from fluoroscopic video frames. The learning algorithm they employ is called Gentle Adaboost to cope with the asymmetric nature of their problem: only around 0.15% of pixels represent the guide-wire. The features they consider are (i) steerable ridge detectors of third and fourth order at several scales, (ii) difference of Gaussians at several couples of scales, (iii) the variance of (ii), (iv) the variance of the image minus the same image in the previous frame. As postprocessing, line segments are obtained by clustering and some of them are selected as being part of the guidewire, based on their rank and orientation.

Saar and Talvik [150] extracted and classified road pavement cracks with a supervised artificial intelligence algorithm. They divide images in 96×96-pixel blocks, which are divided into 12×12-pixel subblocks. Their supervised algorithm considers a neural network with 7 inputs, 4 outputs and 21 hidden layers, and processes entire blocks. The features considered are (i) an indicator of the ratio of subblocks whose mean intensity values is lower than the block one, (ii) horizontal and vertical sums, (iii) median and maximum of the values corresponding to summing column- and row-wise the pixels block, (iv) the summation of the subblock thresholded with a 11×11-pixel convolution mask, both with a horizontal and a vertical bar-shaped kernel. The four outputs correspond to the three types of cracks and the non-crack output.

Becker *et al.* [25] presented an artificial intelligence based algorithm using a classificator based on Gradient Boosting. It is, like AdaBoost, a cascade of weak learners. To focus on local features,

the employed weak learners that are based on convolutions with square-windowed kernels.

Oliveira and Correia [136] presented an unsupervised artificial intelligence method to segment and classify road cracks.They divide the image into 75×75-pixel non-overlapping windows, since this choice offers a good trade-off between low computational requirements and low false positive detections. Their method extracts two features per window, the average intensity of its pixels and their standard deviation, and normalizes them. Different clustering algorithms are employed to label each window and to create a quadratic boundary decision between positive and negative pixels. The clustering technique yielding the best results appears to be the mixture of two Gaussian functions.

Wavefront Propagation

Some methods start with a seed and extract a region iteratively, by checking the neighbouring pixels close to the ones already accepted, and including them or not based on specific criteria. Their expressive power is tightly related to their acceptance criteria, although it tends to be low since these algorithms often disregard global information like the position of pixels. They are often based on local properties (*e.g.* the intensity of the pixel being checked), although can also take into account information related to their neighbourhood (*e.g.* the amount of pixels in the parent branch). They tend to be fast methods, often used in real-time applications. Some of them make use of a user-provided seed point, being thus semi-automatic methods. Of course, the computational complexity depends on the acceptance criteria, which can be very basic or extremely complex.

Wang *et al.* [185] segmented axons from 3D microscopic image stacks. After a preprocessing step, a tracing algorithm provides the segmentation. It leverages three features of the curvilinear objects of interest: (i) their smoothness: the centerline, which is estimated by the change of direction of the eigenvector of the Hessian matrix with the largest eigenvalue, should only have subtle deviations; (ii) their proximity: the euclidean distance between two consecutive pixels should be small; and (iii) their continuity: the intensity along the central line should vary smoothly. To face the situations in which more than one axon convey into the same vertex, a reverse tracing strategy is employed. A similar approach is employed by Xu *et al.* [195] to segment astrocytes from confocal microscope images, in which points are added iteratively one by one, according to a cost function that takes into account a closely related criteria.

Van Rikxoort *et al.* [179] extracted airways from CT scans with a region-based method. Their method uses a wave-propagating strategy that considers a large neighbourhood of 80 pixels and starts at the trachea, which is relatively easy to detect. The voxels are accepted if (i) their intensity is lower than an adaptive threshold, or if (ii) the average intensity of a 3×3×3 cube is below it. A group of voxels is accepted as a segment if it fits a number of constraints that are concerned with its shape, its radius with respect to that of its parents, its angle with respect to the previous segment, etc. All of these constraints are designed to minimize leakage. Tschirren *et al.* [172] presented a similar method in which the threshold was increased until a sudden flood was detected, which was interpreted as a leakage.

Feuerstein *et al.* [65] presented a region-based method in which a wavefront propagation is performed restricted to a local cube-shaped region. Firstly, a large branch is quickly detected and the initial cuboid is placed with the appropriate orientation and size. Within it, a sharpening filter is applied and a wavefront propagation algorithm is executed from the seed point, selecting the voxels greater than a threshold. Afterwards, bifurcations or trifurcations are detected by considering the connected components in the boundary of the cuboid, and new cuboids are set to repeat the process iteratively until no more child branches are found. Lee and Reeves [105] applied a similar idea in their region-based method, in which a cylinder is considered instead of the cuboid. Instead of considering only bifurcations or trifurcations, they allow multiple child branches and limit leaks by considering childs only on the end of the cylinder. Their region growing process adds a new

voxel if (i) it is the 6-connected neighbour of a tagged pixel, (ii) its intensity is below a threshold and (iii) the intensity of at least half of its 26 neighbours is below a threshold. This last constraint is intended to further prevent leakage.

Vlachos and Dermatas [183] presented a region-based method to segment retinal vessels. A simple brightness rule provides some seeding pixels which are the basis of a tracking procedure that ends when a cross-sectional model is violated. This tracking procedure is employed at different scales, and a confidence map is generated for each of them. The confidence map is used to merge them into a single-scale response, which is postprocessed with a median filtering and some morphological operations intended to regularize the output.

Miscellaneous

In the following, a series of methods that do not fall into any of the previous category are introduced. Although they are not widely used, they may provide an original insight towards segmenting curvilinear structures.

Several methods are based on rule merging, which are basic statistical measures usually contrasted with a fixed threshold. The merging process varies greatly, although it is usually implemented by means of conjunctions and disjunctions. Tupin *et al.* [173, 174] used a line-detection rule and a cross-correlation detector between pixels in two regions. Chaudhuri and Samal [49] extracted bridges in multispectral satellite imagery by considering simple constraints such as "a bridge should be a concrete structure", "a bridge divides two water regions", "a bridge should have a constant width", etc. Qian *et al.* [145] extracted vessels with a voxel-wise brightness indicator $b(x)$ and a voxel-wise tightness indicator $v(x)$, to derive a voxel-wise curvilinear-object indicator: $v(x) \cdot b(x)$. We remark that the product operation may be seen as a generalization of the binary *and* operator. Zwiggelaar *et al.* [205] detected spiculated lesions in mammograms using an orientation measure, obtained by maximizing the gray-level summation of pixels within the tube-like line model; and a strength measure, which is the mean gray-level intensities along a line minus the mean gray-level intensities of its surrounding background. Given their nature, all of them employ a strategy based on pixels.

Liu *et al.* [111] introduced a region-based approach to segment cracks in pavement by using the properties of connected components. They used histogram equalization followed by spatial filtering to obtain a rough binary estimation. Their main contribution is the transformation of this intermediate result into a fine-grained segmentation. To do so, they studied the connected components of the rough estimation, including their number of pixels, the area of their circumscribed circle, their *directionality*, etc. Their method is complex and includes a lot of parameters in an attempt to consider as many characteristics as possible.

Bismuth *et al.* presented a method to detect guide-wires. Similarly to [147, 45] they apply voting to find curvilinear structures. In this case, they find a very large number of small segments, that are considered potential chunks of the curvilinear object. Then, each segment contributes to each pixel with a specific weight, and the pixels whose accummulated weight is higher are interpreted as part of a curvilinear object. Besides, their method includes a top-hat transform to preprocess input data and a pruning procedure as postprocessing to refine the final result. Since they use voting, the strategy to provide a segmentation can be considered to be based on shapes.

Bauer *et al.* [24] segmented airways from CT scans with a technique based on graphs. Their method outputs a graph that represents the centerline of the binary curvilinear tree. Firstly, a rough estimation of the graph is given by a rough estimation of the centerlines. Then, it overcompletes the graph by adding more potential centerlines between nodes based on the intensity of the voxels and the orientation of the original centerlines. From this graph, they compute a directed and weighted graph which is processed with a graph-based optimization algorithm to select the final estimation

of the curvilinear object's centerline. Their algorithm does not provide pixel-wise segmentations, but graphs describing the centerlines of the curvilinear objects.

3.4 Metrics and Comparisons

This paper is aimed at studying the strengths and weaknesses of different mathematical techniques to segment different types of curvilinear structures. To do so, we have already classified the algorithms according to the technique they use in Sect. 3.3. In this section, a comparison of the algorithms will be presented according to an appropriate methodology to do so.

Although our goal is to compare the different mathematical approaches, we can only compare objectively the algorithms that employ them. We can assume that a better algorithm probably considers a better mathematical formulation of the problem. This reasoning allows us to infer which mathematical tools tend to be better depending on the specific problem.

3.4.1 On the Comparison of Algorithms

The visual comparison of results is useful as a first indicator, but it is not enough to decide which method is *better*. Actually, the definition of *better* must be formalized, and it may be different depending on the objective of the algorithm. The objective comparison of methods will depend on the performance measure used and on the set of processed images. Once they have been established, statistical tests must be used to provide confidence of the superiority of one method over another.

To compare algorithms, each data sample should be associated with a ground truth, which is the desired result after processing the image. Ideally, experts in the field generate the ground truth, which is often a manual and tedious process. In cases where discrepancy between experts may exist, more than one ground truth may be useful to measure such discrepancy.

In some situations, an edit-based validation may be considered to create, easily and quickly, ground truths [176]. If a method provides results of enough quality, an expert may refine them by iteratively changing specific parts (*e.g.* adding or removing pixels to the ground truth) with the help of specifically designed tools [138].

A *performance measure* is an indicator of the similarity of an estimation with the desired result. It should be defined according to the final task, which is not the segmentation itself but the further extraction of information. Considering a series of performance measures complements our study by showing the same results from different points of view.

Statistical tools, on the other hand, answer the question of whether an algorithm is *better* than another one in the sense that it provides a significantly better statistical value (*e.g.* the mean, the median, etc.) of the considered performance measures. The test in which we are interested is the *Wilcoxon Signed-Rank test* [189], which analyses whether the mean of two statistical populations differs. It assumes that (i) input images are independent and randomly selected and that (ii) the samples are paired (*i.e.* each observation in one population is paired to an observation in the other, which holds in our case since the input image used is the same). We say that a method is *better* than another with confidence α if and only if the appropriate statistical test provides such result with confidence α.

3.4.2 Performance Measures for Curvilinear Object Segmentation

In the considered literature, there appear many performance metrics. Each of them penalizes deviation from the ground truth differently, and so all of them have advantages and shortcomings. The *right* metric depends on how these deviations should be taken into account. All the measures considered, along with their aim and weaknesses, are summarized in Table 3.3.

TABLE 3.3: Metrics used in the literature to measure the performance of a curvilinear structure segmentation algorithm.

Performance Measures		Broad description	Shortcomings
Pixel wise	Centerline based		
Sensitivity	Completeness Branches detected	Amount of positives that are estimated as such.	They are not affected by incorrectly estimating as positive any negative portion.
Specificity		Amount of negatives that are estimated as such.	It is not affected by incorrectly estimating as negative any positive portion.
Precision	Correctness	Relevance of estimated positives.	They can be artificially improved by identifying only the parts that represent a positive with more probability.
Accuracy		Amount of correct estimations.	Class skewness may induce an undesirable behaviour.
F_1-score		Harmonic mean of Sensitivity and Precision.	It is a complex and non-commutative measure.
	Root Mean Square	Deviation of estimated centerlines.	It does not consider unmatched estimated segments or multiple segments matched to the same portion of ground truth.

Pixel-Wise Measures contrast the estimation against the ground truth in a pixel-wise fashion. We consider several widely accepted performance metrics based on the amount of True Positives and True Negatives (correctly labelled pixels as either belonging to the object of interest or not) and False Positives and False Negatives (wrongly classified pixels, where false positives are the ones estimated positives but actual negatives, and vice versa). False Positives are also known as type I errors, whereas False Negatives are known as type II errors. These measures are the sensitivity (also known as recall), the specificity, the accuracy and the precision:

$$\text{Sen} = \frac{TP}{TP + FN}, \qquad \qquad \text{Spe} = \frac{TN}{TN + FP},$$
$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}, \qquad \text{Pre} = \frac{TP}{TP + FP}.$$

Given that each measure captures a different approach of the *similarity* between the segmentation and the desired result, all of them present shortcomings. Thus, in addition to visual inspection, a combination of them is required towards infering veracious conclusions. For instance, the sensitivity and the specificity are not affected by type I and type II errors, respectively, whereas accuracy is not a faithful representation when one class is much more common than the other. The Precision, which measures the relevance of the estimated positives, can be artificially increased by relabelling as estimated negatives those pixels lacking a strong evidence of being positives. The F_1-score tries to capture both types of errors, FP and FN. In doing so, it assigns them the same weight, which may not be the best approach. Although it can be generalized with the F_β-score (the weigthed harmonic mean with $w_{\text{Sen}} = \frac{\beta^2}{1+\beta^2}$, $w_{\text{Pre}} = \frac{1}{1+\beta^2}$, such weights are still necessarily fixed a priori. Besides, the F_1-score changes if the estimation is exchanged with the ground truth, which contrast with the idea of it being a measure of similarity.

Other approaches evaluate the performance based on a higher level description of the curvilinear segmented objects, like the *Centerline-Based Metrics*. Sometimes the representation of structures is vectorized—so the centerlines are explicitly included—, but they have to be computed when the estimations or ground truths are given as pixel-wise masks. This computation, which can be implemented with the fast marching algorithm [113], may distort the metric. Besides, the metric may be further twisted when the estimation and ground truth's centerlines are matched. The matching process splits both networks into small segments, and assigns each segment of the segmentation to its *best counterpart* on the ground truth, if there is any [188]. We understand by a correctly identified centerlines those derived from the estimation that have been matched to a centerline from the ground truth.

The *Completeness* [188] is concerned with how much of the object has been identified. It is defined as the ratio between the total length of the correctly identified centerlines, and the total length of the centerlines of the ground truth.

The *Correctness* [188] is defined as the total length of the correctly identified centerlines divided by the total length of the centerlines of the estimation.

Another high-level measure is the *Ratio of Branches Detected* [113]. The ground truth is divided into branches, each one having a different, uninterrupted centerline. A branch is considered as detected if is matched with a correctly identified centerline larger than a predefined threshold. Then, we can define the ratio of branches detected as the ratio between the amount of branches detected and the total number of branches.

Finally, the *Root Mean Square* (RMS) [188] is a measure of the goodness of matching the estimated centerline network to the ground truth's network. The RMS measures the average distance between the correctly identified centerlines and their corresponding segments in the ground truth:

$$\text{RMS} = \sqrt{\frac{1}{l} \sum_{i=1}^{l} \text{d}\left(\text{segm}_i, \text{match}(\text{segm}_i)\right)^2}, \tag{3.1}$$

where i ranges over all matched segments, and $\text{d}\left(\text{segm}_i, \text{match}(\text{segm}_i)\right)$ is the distance between the centerline of the i-th matched segment, segm_i, and the centerline of the ground truth segment with which it has been matched, $\text{match}(\text{segm}_i)$. We remark that it ignores the estimated segments that have not been matched with any chunk of ground truth, and it is not affected if several portions of the segmentation are assigned to the same part of the ground truth.

The choice of the best performance measure should be based on the final goal of the segmentation. For instance, extracting the tortuosity of vessels towards a computer-aided diagnosis may require a pixel-wise metric with high accuracy, whereas updating a map from the possible location of roads may be fitted by a centerline-based metric that admits false positives. This choice potentially affects which is the best technique to approach the problem, since as previously said the notion of *best* depends on the measure with which the performance is evaluated.

3.4.3   Datasets

In this section, we present the most representative benchmarks for each application along with different segmentation methods by different authors. Specifically, we have selected the most widespread benchmarks to cover as many segmentation strategies and, potentially, the leading algorithms.

They are compared with the most used metrics, which depend on the application, and their classification. When available, the computation time required to execute each segmentation is also provided. Unless otherwise specified, the time has been obtained using a modern computer at the original year of publication, which is also shown.

Vessel Segmentation in Retinal Angiography

Vessel segmentation in retinal images is a popular topic that has been addressed with a wide variety of techniques. The DRIVE dataset [162] contains 40 colour, medium-sized images of mostly healthy patients, along with two manual segmentations. One of them is used to compute the metrics, whereas the other one—the 2nd ground truth in Tab. 3.4—is used to measure the variation between experts. Usually, the methods are compared with pixel-wise measures, which are well suited in the case of low and medium quality images. The metrics in Tab. 3.4 have been obtained as the average for the 20 test samples, which are the ones having a second expert segmentations. The mask representing the Field Of View (FOV) has been applied to remove background pixels from both the ground truth and the estimations.

Another benchmark for retinal vessel segmentation is the STARE dataset [83]. It contains 397 images, 20 of them including two manual segmentations. These images show a greater amount of pathologies and artifacts. Since there is no mask for the Field Of View (FOV) included in the dataset,

TABLE 3.4: Results of the retinal vessel segmentation benchmark DRIVE [162]. The best automatic method according to the mean of each metric appears in bold. All the others methods are compared against it with the Unilateral Wilcoxon Signed-Rank ($\alpha = 0.05$). The superscripts indicate whether $\sim$ the results are *not* statistically better/worse, $\dagger$ the results are statistically worst, or $\ddagger$ the results are statistically better (in the case of human-assisted segmentations). No superscript indicates that the individual segmentations were not available.

Year	Method	Sensitivity	Specificity	Accuracy	F1 Score	Classification	Based on	Time per sample
-	2nd GT	$0.7760^\sim$	$0.9725^\dagger$	$0.9473^\ddagger$	$0.7881^\ddagger$	Human segmentation	-	~ 3 h
2001	Zana and Klein [199]	$0.6695^\dagger$	**0.9831**	$0.9431^\sim$	$0.7475^\dagger$	Grayscale morphology	Pixels	~ 3 min
2004	Staal *et al.* [162]	$0.7194^\dagger$	$0.9773^\dagger$	$0.9442^\sim$	$0.7644^\dagger$	Eigenvalue of Hessian	Shape	~ 15 min
2006	Soares *et al.* [161]	$0.6735^\dagger$	$0.9822^\sim$	$0.9426^\sim$	$0.7483^\dagger$	Learning model	AI	~ 3 min
2006	Mendonça and Campilho [125]	0.7344	0.9764	0.9452	-	Grayscale morphology	Shape	~ 2.5 min
2007	Wang *et al.* [186]	0.780	0.978	-	-	Parameter fitting	Shape	~ 5 min
2009	Al-Diri *et al.* [58]	0.7282	0.9551	-	-	Active contours	Edges	~ 11 min
2010	Vlachos and Dermatas [183]	0.747	0.955	0.929	-	Wavefront propagation	Region	Unknown.
2012	Fraz *et al.* [68]	0.7152	0.9769	0.9430	-	Grayscale morphology	Shape	~ 100 s
2015	Azzopardi *et al.* [11]	**0.7618**	$0.9716^\dagger$	**0.9447**	**0.7758**	Linear Filtering	Shape	~ 5 s
2015	Bibiloni *et al.* [33]	$0.7490^\sim$	$0.9674^\dagger$	$0.9393^\dagger$	$0.7568^\dagger$	Fuzzy morphology	Region	~ 35 s
2015	Roychowdhury *et al.* [149]	$0.6820^\dagger$	$0.9729^\dagger$	$0.9355^\dagger$	$0.7280^\dagger$	Grayscale morphology	Region	~ 4 s

those pixels have not been removed,[1] so the values of Specificity and Accuracy are optimistic when compared to those of the DRIVE benchmark. Table 3.5 contains the average of the most relevant metrics using the 20 images with two manual segmentations.

TABLE 3.5: Results of the retinal vessel segmentation benchmark STARE [83]. The results of the statistical tests are reflected in the superscripts as in Table 3.4.

Year	Method	Sensitivity	Specificity	Accuracy	F1 Score	Classification	Based on	Time per sample
-	2nd GT	$0.8951^\ddagger$	$0.9562^\dagger$	$0.9522^\dagger$	$0.7401^\sim$	Human segmentation	-	~ 3 h
2001	Hoover *et al.* [83]	$0.6733^\dagger$	$0.9692^\dagger$	$0.9462^\dagger$	$0.6506^\dagger$	Matched Filtering	Shape	~ 5 min
2006	Mendonça and Campilho [125]	0.6996	0.9730	0.9440	-	Grayscale morphology	Shape	~ 3 min
2006	Soares *et al.* [161]	$0.6632^\dagger$	$0.9811^\dagger$	$0.9566^\dagger$	$0.6982^\sim$	Learning model	AI	~ 3 min
2007	Salem *et al.* [151]	0.8215	0.9750	-	-	Learning model	User	Unknown.
2007	Wang *et al.* [186]	0.752	0.980	-	-	Parameter fitting	Shape	~ 5 min
2009	Al-Diri *et al.* [58]	0.7521	0.9681	-	-	Active contours	Edges	~ 11 min
2012	Fraz *et al.* [68]	0.7311	0.9680	0.9442	-	Grayscale morphology	Shape	~ 100 s
2015	Azzopardi *et al.* [11]	**0.8192**	$0.9653^\dagger$	$0.9544^\dagger$	**0.7315**	Linear filtering	Shape	~ 5 s
2015	Bibiloni *et al.* [33]	$0.7500^\dagger$	$0.9704^\dagger$	$0.9539^\dagger$	$0.7093^\dagger$	Fuzzy morphology	Region	~ 35 s
2015	Roychowdury *et al.* [149]	$0.6761^\dagger$	**0.9852**	**0.9618**	$0.7264^\sim$	Grayscale morphology	Region	~ 4 s

Airway Extraction in CT Scans

A short number of years ago, the database EXACT [113] was collected to build a framework upon which algorithms facing airway tree segmentation could be compared. The high quality of the input images, which are three-dimensional representations of the chest acquired with different multi-slice Computed Tomography (CT) scans, enables the use of high-level metrics. Some regions are not included in the calculation of the metrics, like the regions labelled as "unknown" in the ground truth or the trachea. The most relevant results are shown in Table 3.6.

We remark that a great number of methods employ a wavefront propagation technique possibly because the framework only accepted segmentations with a single connected component.

Road Detection

Although lots of methods in the literature deal with road segmentation, they are barely compared with each other. This is probably due to the huge variety of images that may be used: they can be either aerial or satellite images, represent urban or rural areas, have distinct spatial resolution, collect information from different spectral bands, etc. Mayer *et al.* [119] proposed a comparison with 3 aerial images and 3 satellite images, along with some measures based on centerlines: correctness,

---

[1] Estimating the FOV *a posteriori* could affect the fairness of the comparison, since different methods could consider slightly different FOV. We have preferred not to include any while remarking that the performance metrics may be biased.

TABLE 3.6: Results of the airway extraction benchmark EXACT'09 [113]. No statistical tests were performed since only the average of the metrics were known.

Year	Method	Branches detected	Completeness	Correctness	Classification	Based on	Time per sample
2009	Fetita *et al.* [64]	0.628	0.559	0.9804	Grayscale morphology	Region	∼ 5 min
2009	Feuerstein *et al.* [65]	0.765	0.733	0.8444	Wavefront propagation	Region	∼ 5 min
2009	Lo *et al.* [112]	0.598	0.540	**0.9989**	Learning model	Region	∼ 90 min
2009	Lee and Reeves *et al.* [105]	0.324	0.281	**0.9989**	Wavefront propagation	Region	∼ 30 s
2009	Born *et al.* [40]	0.417	0.345	0.9959	Wavefront propagation	User	∼ 25 s
2009	Bauer *et al.* [23]	0.630	0.584	0.9856	Gradient vector flow	Region	∼ 6 min
2009	Tschirren *et al.* [172]	0.631	0.589	0.9881	Wavefront propagation	User	∼ 60 min
2009	Van Rikxoort *et al.* [179]	0.672	0.570	0.9273	Wavefront propagation	Region	∼ 10 s
2012	Lo *et al.* [113]	**0.843**	**0.788**	0.9878	Combination of other methods		Unknown.
2012	Wang *et al.* [187]	0.507	0.456	0.9765	Statistical model	Shape	∼ 45 s
2013	Xu *et al.* [196]	0.517	0.445	0.9915	Grayscale morphology	Region	∼ 20 min
2015	Bauer *et al.* [24]	0.711	0.716	0.9025	Graph	-	10 min

completeness and RMS. The low number of images and methods makes the comparison incomplete. Although some methods perform acceptable well, more work is needed to infer meaningful conclusions regarding the most suitable methodology.

3.4.4 Other Comparisons

Other comparisons are now presented to convey more information regarding the adequacy of the studied mathematical tools towards curvilinear object segmentation. Only a few reliable benchmarks are available in the literature. Thus, lower-quality comparisons can help to complement such information. This includes quantitative comparisons between a few methods or with a very small amount of samples, several implementations by the same author or even qualitative comparisons.

We remark that these results do not have the same importance than publicly available benchmarks, and this should be reflected when deriving conclusions from them.

Wire Detection in Low-Flight Imagery

Low and high altitude rotorcrafts provide low-resolution video from which wires can potentially be detected. Tanović [169] implemented three different methods and compared them:

(W1) A parametric, edge-based procedure based on the work by Candamo and Goldgod [43]. It employs the Canny edge detector, a parametric line fitting over the edge map and some postprocessing to increase the accuracy.

(W2) Following the same line of research, it is introduced an Edge-Based procedure that leverages the Hough transform. It substitutes the parametric line fitting and further postprocessing by a windowed Hough transform.

(W3) The last implementation corresponds to a modification of the Passive Obstacle Detection System (PODS) [152], already classified as a $(0, 1)$-tensor based method.

The implementation of the algorithms, used to compute the performance measures, were based on the original publications. Besides, the dataset used corresponds to images taken with a static rotorcraft subjected to camera movements, whereas low-altitude wire detection commonly uses a static camera in a moving rotorcraft.

The performance measures used are based on centerlines, since they tolerate small shift and orientation deviations, and only take into account the amount of lines detected or not, ignoring their length. The (W3) algorithm provides a Completeness of 77%, compared to a 72-74% provided by the other algorithms. (W2) yields the greatest Correctness (66%) in comparison with (W1) (61%) and (W3) (55%). Besides, (W2) is the faster algorithm (1 s/sample), being (W3) the slowest (3 s/sample).

Human Wrinkle Detection

Although age estimation from pictures has been widely studied, only a few methods explicitly segment the wrinkles. Batool *et al.* presented two methods and compared them. The first method, (H1) [20], is a method based on Marked Points Processes; whereas the second one, (H2) [19], is based on enhancement with Gabor filters.

The metrics used for comparison are pixel-wise, allowing 3 pixels of tolerance between the estimation and the ground truth. They do not take into account the centerlines or their orientation. (H1) obtains a good sensitivity ($\sim 85\%$), and offers good specificity ($\sim 95 - 98\%$), whereas (H2) provides an average sensitivity ($\sim 40 - 70\%$) but a very high specificity ($\sim 99.5\%$). (H2) is much faster (9 s/sample instead in contrast with 65 s/sample).

3.5   Conclusions

Lastly, we discuss the results shown in previous sections and conclude with the limitations of this survey. The discussion intends to unify the most useful information and present it so practitioners can extract useful and specific insights.

3.5.1   Discussion

First, regarding the categories in Fig. 3.6, there is neither a best algorithm to face all the situations nor a *best mathematical approach* to do so. Deciding which methodology is best may depend on (i) the specific model of curvilinear objects—inferred both by the application and the specific database used—, (ii) the evaluation metric and (iii) the computational requirements that will bound our algorithm. However, when inspectioning the results, we observe that some simple techniques achieve very good results. Linear filtering and mathematical morphology [11, 149] are two techniques with which methods can achieve good performance according to all metrics simultaneously. Specially important are these techniques when applications require quick responses or real-time interaction. On the other hand, more complex mathematical tools can be used to pursue better performance when the computational cost is not a constraint [24] or when a very complete model for curvilinear objects is explicitly introduced [165].

Second, with respect to the taxonomy in Fig. 3.7, the *strategy with which the best segmentations are performed* is usually region-based. It is clear that a pixel-based strategy is inadequate under certain circumstances, like detecting roads occluded by trees or shadows [177]. The methods based on edges and shapes can hardly leverage the global structure of the object, although some successful instances have been published [11]. Although region-based algorithms can be seen as the most powerful, they tend to cause large leakages [144] and are too sensitive to small threshold variations [179]. Several approaches have been considered to deal with these problems, such as detecting leaks as large increments when slightly increasing a threshold (the so-called explosion control [112, 172]) or by leveraging the specific model for curvilinear structures (for instance, forcing the diameter of branches to be non-increasing [179] or iteratively limiting the pixels in which region-growing is executed [105], among others).

Third, a series of observations may help to select the methodology that best fits their needs. This, which is one of the goals of this survey, is related with several independent aspects of the design of segmentation algorithms:

Simple Mathematical Models. The use of a model with few parameters tends to avoid overfitting [186] and can face high levels of noise. For instance, guide-wire extraction may handle discontinuities by leveraging the fact that there is only one curvilinear object, which can be modelled as a 2nd or 3rd order curve [140]. On the other hand, complex methods may largely increase the computational requirements [26] and tend to overfit the training data. Of course, the model should be appropriate, since violations of the model lead to misbehaviour [63].

Multiscale. Different approaches are considered to deal with curvilinear structures that have different sizes.

Most algorithms use a multi-step approach: they consider variations of the same operators and aggregate the results afterwards. When spatial derivatives are computed by convolving the image with the derivative of a Gaussian shape, multiple Gaussian shapes can be used by considering different standard deviations [154]. Linear filters, besides being rotated several times, can be enlarged or shrinked [11].

Other strategies to handle multiple-width objects use a single operation. The Gradient Vector Flow [194] is an edge-preserving diffusion of the traditional gradient, specially designed to capture concavities and to present a large capture range. It allows the detection of centerlines of curvilinear objects regardless of their width [22]. A clustering algorithm may also be used to consider several widths at once [151]. However, this approach implicitly assumes a certain width distribution when defining the distance between pixels and cluster centroids.

Closely Placed Curvilinear Objects. Some types of images present tubular-shaped objects near others along a certain path. This situation is hard to handle. For instance, the approaches that employ multiple-scale filters tend to present, in these regions, a response dominated by the large scale filters [84], and the algorithms that smooth the images with Gaussian functions blur such regions [191]. The Dual-Gaussian or Bi-Gaussian function [191, 110] has been introduced as a kernel to face this specific problem. The use of specific active contours may also be used to face this situation [58].

User-assisted methods. Although semi-automatic methods are considered, they do not tend to produce better results. The algorithms in which the user provides a small number of seed points provide similar results to the fully automatic ones [40], maybe because this operation can be robustly implemented. In contrast, the algorithms in which the user constantly supervises the algorithm tend to provide really good correctness but average completeness [112, 172].

Artificial Intelligence techniques. Similarly, the results provided by machine learning techniques are comparable but not better than the other approaches in terms of performance measures [161]. We remark, however, that their usually costly learning stage and their lack of interpretability can be compensated with the short time needed to process new samples.

Another important aspect of algorithms, barely mentioned in this survey, is their computational requirements. Some strategies have been considered to decrease the computational requirements of algorithms. A few mathematical techniques have been developed with this objective in mind: instead of considering all segments in each image, as does the original Hough transform, they consider only a subset of them by dividing the image into grids and subgrids. Both the Beamlet transform [59] and the quad-tree structure [186] use this approach. An alternative to diminishing computational requirements while maintaining robustness is the use of randomized algorithms. They can be employed for parameter fitting [82, 18] or to fit complex statistical models [165].

Finally, we highlight two limitations of this discussion. On the one hand, the performance measures are just *indicators*. Segmentation is usually a single step of a larger system, and the best segmentation methodology is the one that provides the best results in such system when embedded into it. Given the cross-application nature of this survey, the discussion is based on performance measures, so slight deviations can be found when implementing end-to-end systems. On the other hand, this discussion is limited by the availability of segmentation algorithms and datasets. More information is needed to draw definitive conclusions. Specifically, more widely accepted benchmarks in applications such as road extraction from aerial images, and more methods employing statistical models and other underexplored techniques. Besides, statistical tests should be considered a requirement to provide fair comparisons between different segmentation algorithms.

3.5.2   Limitations of this Survey

Some limitations have been already mentioned in Sect. 3.5.1. Specifically, the limited amount of methods and datasets, and the suitability of performance metrics to reflect the correctness of algorithms. Although a lot of work has been done defining and using performance measures, more connections should be established between the final goal of the segmentation (*i.e.* the end-to-end system) and how to measure the correctness of the segmentation. This step is often forgotten by the methods that face a particular problem, but it is of paramount importance in order to use them in real applications.

Besides, the conclusions we draw regarding how well-fitted are certain mathematical techniques or certain strategies to segment objects may be inaccurate. We have compared algorithms that employ them and, then, we have assumed that the best algorithms faithfully represent their capacities. Although it seems coherent and is unavoidable, this assumption should be explicited.

Finally, to further improve the state of the art in curvilinear object detection, more datasets are needed to compare algorithms designed for some application. Examples of such applications are road detection, guide-wire tracking or pavement assessment. The latter has been addressed and some datasets are publicly available [203], but they are still not popular enough to compare different approaches. Of course, the publication of more segmentation algorithms would also complete the cross-application state of the art of curvilinear object segmentation.

Chapter 4

Curvilinear Object Detection with Fuzzy Mathematical Morphology

As discussed in Sect. 3.1, curvilinear object detection is the common denominator of many applications. Some illustrative examples are road detection from aerial or satellite images, human airways from volumetric 3D scans or vascular structures in eye-fundus images. In this chapter, we propose two general-purpose curvilinear object detectors that may serve as building blocks for application-specific systems. To do so, we employ the fuzzy mathematical morphology operators, defined in Sect. 2.4, due to their robustness with respect to uncertainty and noise, and the trade-off they offer between expressive power and computational requirements. The extraction of curvilinear features is based on, respectively, the fuzzy hit-or-miss transform and the fuzzy top-hat transform. To demonstrate their potential to extract curvilinear objects, the will be employed in Chapter 5 to detect the curvilinear objects that vessels represent in retinal images.

Sect. 4.1 introduces the problem and the current state of the art of general-purpose curvilinear object detectors. Then, we introduce our general-purpose algorithms in Sect. 4.2 and 4.3. In Sect. 4.4, we compare these two algorithms with the best alternatives from the related literature. Our aim is to highlight the characteristic of each of the algorithms to ease the problem of choosing the one that best fits into each specific application. We conclude discussing such strengths, weaknesses and limitations of our two algorithms in Sect. 4.5.

4.1 Introduction

Curvilinear structures are shapes found in a great variety of images. Instances of them include hair in dermoscopic imagery, cracks in pavement and bronchi in volumetric computed tomography scans. An application-grounded, general definition of curvilinear structures has already been provided in Sect. 3.2, where they are described as a thin and long region of connected pixels, whose width and intensity levels vary smoothly and whose cross-section profile presents pixels with significantly different intensities than the ones surrounding the curvilinear object. Curvilinear objects are not contours (*i.e.* the limits between two regions), but tubular-shaped objects located in the foreground of a uniform or non-uniform region.

General-purpose curvilinear object detectors have an important role to play. A great number of works have been dedicated to just one specific application. See, for instance, the review by Fraz *et al.* [69] of retinal vessel segmentation methods, or the survey by Lesage *et al.* [108] of 3D vessel lumen extraction algorithms [113]. Other examples where extensive work has been done include road pavement assessment through crack detection [46], road extraction algorithms [124], palmprint line enhancement [97] or fingerprint classification [197]. These categories represent a vast amount of algorithms that, essentially, pursue a similar goal. In spite of the amount of works that segment curvilinear objects, few of them consider generic curvilinear objects or different situations where they can be found. Instead of being used across different applications, the same techniques

are rediscovered and fine-grained by researchers facing similar situations [32]. General-purpose curvilinear object detectors can be of great help by providing efficient segmentations at the cost of undemanding customizations.

Krylov and Nelson [99] introduced a generic curvilinear object extraction algorithm. They divide the image into an overlapping grid of square regions and use the finite Radon transform to find segments in them. Afterwards, a pixel-wise energy term is computed and weak, isolated candidates are removed by thresholding such energy indicator. This is done taking into account the segmented regions themselves, by means of a template matching procedure; and how they are related, with the orientation they present and the distance between them. Their output is, due to the nature of the Radon transform, a set of segments described by their position, orientation and strength.

Bai *et al.* [14] presented a general-purpose curvilinear structure segmentation algorithm. It is based on the grayscale hit-or-miss transform, which uses flat structuring elements. They employ a bar-like structuring element at different scales and different orientations, obtaining several masks. These masks are aggregated into a single binary result, in which a pixel is selected provided that it had a positive value in any of the masks. Therefore, this method basically provides a binary result, which is then postprocessed to capture the skeleton of the curvilinear structures.

Heijmans *et al.* [81] introduced path openings and closings. They are morphological operators that are especially designed to detect curvilinear structures of at least L pixels connected according to a directional adjacency matrix (that is, restricting the possible connected pixels to the upper and right pixels in their 8-neighbourhood). They have been adapted to detect incomplete paths and to be used in high-dimensional settings [53, 168]. Path openings and closings have been used to detect vessels in retinal images [160] and to detect roads and rivers from satellite imagery [94, 177]. Merveille *et al.* [128] used path operators to design an operator called Ranking the Orientation Responses of Path Operators (RORPO). This new operator was developed to distinguish curvilinear objects from blob-like and planar structures in 3D imagery, leveraging the fact that curvilinear objects output high values at a specific orientation (considered by the directional adjacency matrices). Recently, a version of the RORPO algorithm for 2D imagery was proposed in [127].

The goal of this chapter is to present, study and compare two simple yet powerful general-purpose curvilinear detectors. These operators, introduced in Sect. 4.2 and 4.3, output a grayscale image that represents the pixels' degree of belonging to a curvilinear object. They are designed to be easily embedded into bigger systems but, at the same time, to be adaptable to a wide range of specific situations. We aim at bridging the gap between different applications sharing the common denominator of curvilinear object extraction or segmentation. As it is demonstrated in Sect. 4.4, the two proposed curvilinear detectors are able to locate black and white curvilinear objects of varying width while ignoring noise and blob-shaped structures. These detectors are able to differentiate curvilinear structures based on their strength, and succeed at extracting thin curvilinear objects, detecting bifurcations and crossovers and suppressing noise.

4.2   Curvilinear Object Detector based on the Fuzzy Hit-or-Miss

The first detector we present, inspired on the work by Bai *et al.*, is based on the fuzzy hit-or-miss operator. As explained in Sect. 2.4, the fuzzy hit-or-miss operator enforces the appearance of a predefined shape and the absence of another predefined shape. Such shapes are (i) a bar-shaped, with a certain width and orientation, and with a Gaussian cross-section profile to fit curvilinear objects with gradual borders; and (ii) the complementary of (i). Due to its nature, besides processing grayscale images, it outputs a grayscale image as a pixel-wise indicator of the pixel's degree of belonging to a curvilinear object. This soft response reduces the amount of noise introduced by the algorithm and it is suitable for further postprocessing in order to tailor the result to each specific application.

This detector employs several structuring elements. Each of them has the same shape, a line-like profile with a Gaussian decayment, and differ in terms of their size and their orientation. This choice is made based on the behaviour of the hit-or-miss: similarly to [14], the hit-or-miss operator is adequate to extract curvilinear objects in a given direction. Besides, the Gaussian decayment profile, is based on the hypothesis that curvilinear objects have, approximately, a Gaussian cross-section profile, a valid assumption in several applications, as shown in Table 3.2.

More formally, we consider pairs of structuring elements for a range of sizes $s_1, \dots, s_n$ and orientations $\alpha_1, \dots, \alpha_m$. For all i such that $1 \leqslant i \leqslant n$, s_i indicates the side, in pixels, of the square-shaped structuring element; and for all j such that $1 \leqslant j \leqslant m$, α_j measures, in degrees, the angle between the horizontal and the centerline of the line. Each structuring element is mathematically described as follows. Let us consider a line that crosses the point $(0,0)$ with orientation α. Then, the distance of any point of the plane (x,y) to such line is $d_\alpha(x,y) = \frac{tg(\alpha) \cdot x - y}{\sqrt{tg(\alpha)^2 + 1}}$. To provide a Gaussian-like decayment to a bar with orientation α, we consider our structuring elements of size s as:

$$B_1^{(s,\alpha)} : \{1, 2, \dots, s\}^2 \longrightarrow [0,1], \qquad B_2^{(s,\alpha)} = 1 - B_1^{(s,\alpha)},$$

$$(x,y) \longmapsto e^{-\left(\frac{d_\alpha(x,y)}{2\sigma}\right)^2},$$

where the variance has been manually adjusted to $\sigma = s/7$ so that the bar-like template properly fills the structuring element's area. Examples of such shapes can be seen in Fig. 4.1.
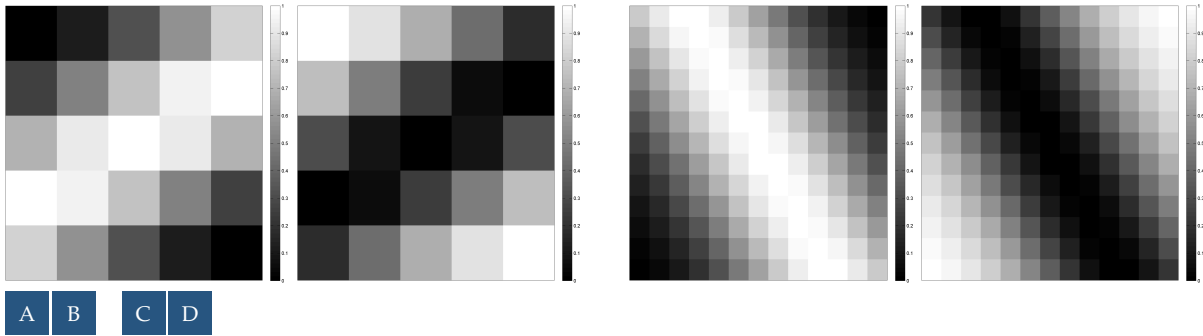

A B C D

FIGURE 4.1: Pairs of structuring elements for the hit-or-miss transform. (A) and (B) $s_1 = 5$ and $\alpha_1 = 30°$. (C) and (D) $s_2 = 13$ and $\alpha_2 = 120°$.

We remark that the structuring element $B_1^{(s,\alpha)}$ is normalized (*i.e.* a linear re-mapping is applied so that its minimum is 0 and its maximum is 1). This slightly changes its values but, as shown below in Fig. 4.6, represents a great advantage since it helps to ensure that the fuzzy hit-or-miss operators outputs zeros in flat regions, which is shown in Proposition 4.1.

Proposition 4.1. — *Let C be a fuzzy conjunction, let I be a fuzzy implication function, let N be a strong fuzzy negation and let A, B_1, B_2, be grayscale images. Let us assume that (i) A is constant in a neighbourhood of x_0, given by $T_{x_0}(d_{B_1} \cup d_{B_2})$, (ii) B_1 and B_2 have, each, at least one pixel valued 1, (iii) I fulfills* (NP)*, and (iv) C and N fulfill* (NC)*.*

Then, it holds that $\mathcal{FHM}_{C,I,N}(A, B_1, B_2)(x_0) = 0$.

Proof. In flat areas, we can assume that A is constant in a local neighbourhood. In particular, by hypothesis we have that $A(x_0 + \delta) = k \in [0,1]$ for any $\delta \in d_{B_1} \cup d_{B_2}$. By definition, we have

$$\mathcal{E}_I(A, B_1)(x_0) = \inf_{x \in d_A \cap T_{x_0}(d_{B_1})} I\big(B_1(x - x_0), A(x)\big) = \inf_{x \in T_{-x_0}(d_A) \cap d_{B_1}} I\big(B_1(x), A(x + x_0)\big)$$

$$= \inf_{x \in T_{-x_0}(d_A) \cap d_{B_1}} I\big(B_1(x), k\big).$$

Since I is decreasing in the first variable and $B_1(x') = 1$ in some location x', $\mathcal{E}_I(A, B_1) = I(1, k) = k$, where the last equality holds since I fulfills the (NP) property. Similarly, $\mathcal{E}_I(N(A), B_2) = N(k)$ and $\mathcal{F}\mathcal{H}\mathcal{M}_{C,I,N}(A, B_1, B_2) = C(k, N(k)) = 0$ due to the (NC) principle (see Prop. 2.14).   □

  Although the hypotheses are quite technical, a variety of common fuzzy operators fulfill them, as can be seen in Tab. 2.1 (page 11).

  The workflow of the curvilinear object detector based on the fuzzy hit-or-miss transform is illustrated in Fig. 4.2. In order to cope with the differences in size and orientation of curvilinear structures, several fuzzy hit-or-miss operators are computed. They are then aggregated into a single output that estimates, for each pixel, its degree of belonging to a curvilinear object. Since the detector is designed to extract light curvilinear objects, images that contain dark curvilinear objects are inverted (*i.e.* $A'(x, y) = 1 - A(x, y)$) before being processed with the detector.
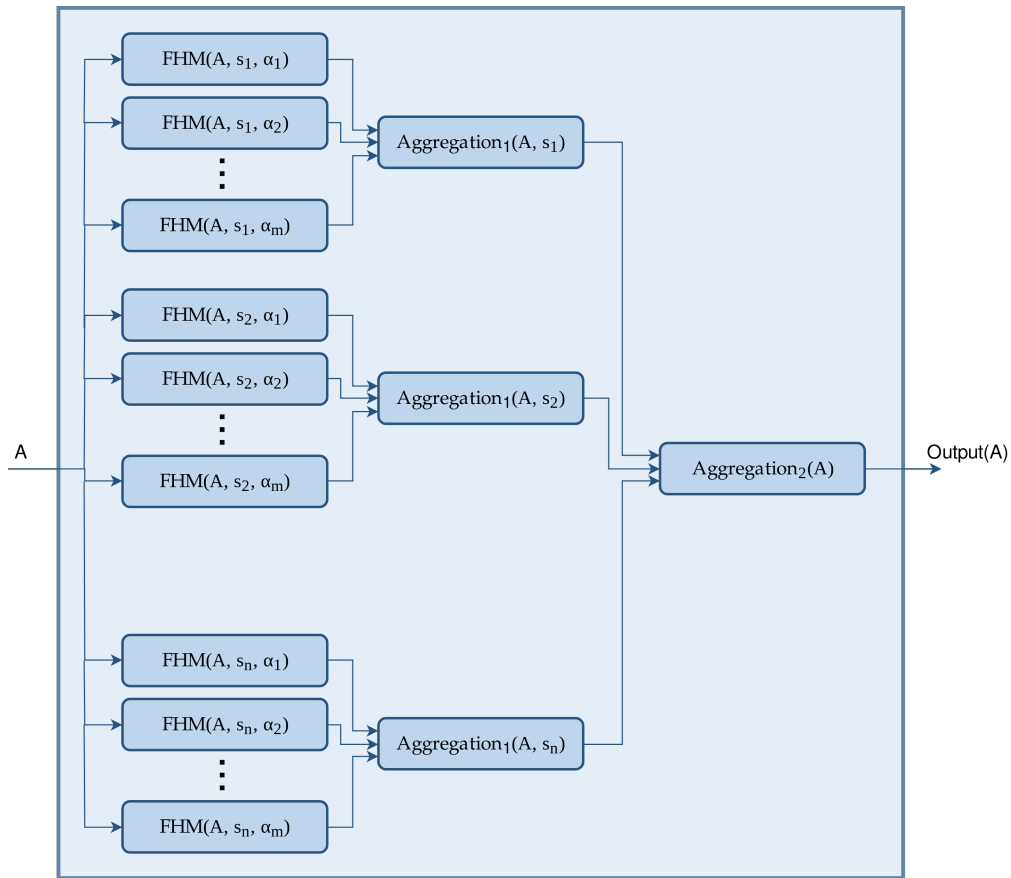

FIGURE 4.2: Workflow of the hit-or-miss curvilinear object detector applied at scales $\{s_1, \dots, s_n\}$ and orientations $\{\alpha_1, \dots, \alpha_m\}$. We remark that $\text{FHM}(A, s_i, \alpha_j)$ represents, in short, $\mathcal{F}\mathcal{H}\mathcal{M}_{C,I,N}(A, B_1^{(s_i, \alpha_j)}, B_2^{(s_i, \alpha_j)})$, where C, I, N should be specified, $B_1^{(s_i, \alpha_j)}$ is a $s_i \times s_i$-pixel structuring element with a bar-like shape at the orientation α_j, and $B_2^{(s_i, \alpha_j)} = 1 - B_1^{(s_i, \alpha_j)}$ (see Fig. 4.1).

  The curvilinear detector based on the fuzzy hit-or-miss operator is formally specified as follows. Let $\text{Agg}_1, \text{Agg}_2$ be two aggregation functions from $[0, 1]^r$ to $[0, 1]$, for an arbitrary dimension r. Let C, I, N be, respectively, a fuzzy conjunction, a fuzzy implication function and a strong fuzzy negation. Let A be a grayscale image. Then, the *curvilinear object detector based on the fuzzy hit-or-miss operator*, $\mathcal{D}_1$, is defined as:

$$\mathcal{D}_1(A) = \underset{s \in \{s_1, \dots, s_n\}}{\text{Agg}_2} \left\{ \underset{\alpha \in \{\alpha_1, \dots, \alpha_m\}}{\text{Agg}_1} \left\{ \mathcal{F}\mathcal{H}\mathcal{M}_{C,I,N}(A, B_1^{(s, \alpha)}, B_2^{(s, \alpha)}) \right\} \right\}. \tag{4.1}$$

where $s_1, \ldots, s_n$ and $\alpha_1, \ldots, \alpha_m$ denote the sizes and orientations of the structuring elements $B_1^{(s,\alpha)}$ and $B_2^{(s,\alpha)}$, which are specified as before. We remark that Agg_1 and Agg_2 strongly influence the behaviour of the detector, and their effects are discussed below, along with a visual representation in Figure 4.5.

Unless otherwise stated, the parameters used when instantiating this detector are as follows. The hit-or-miss operator is computed along 6 orientations (*i.e.* $\alpha_k = \frac{180°}{6}k$ for $k = 1, \ldots, 6$), with structuring elements of sizes 5×5, 9×9 and 13×13, the Łukasiewicz t-norm $T_\mathbf{L}$ as the conjunction C, the Łukasiewicz implication $I_\mathbf{L}$ as the implication I, the standard negation as N and the probabilistic sum $S_\mathbf{P}$ as both aggregation functions, Agg_1 and Agg_2. In the rest of the section we study its behaviour with respect to these parameters. This analysis (i) motivates the election of the aforementioned parameters and (ii) shows the practitioner the range of behaviours that this detector can offer.

Different parameters can be customized. First, the sizes $s_1, \ldots, s_n$ determine the thickness of the curvilinear objects of interest. The aggregation across scales ensures that we capture curvilinear objects whose width is within a certain range. In Fig. 4.3, the method is executed with different size ranges. As we can observe, the width of the detected curvilinear objects corrrespond to the size of the structuring elements being used in each case. Moreover, the shape of the structuring elements could be modified to enhance other cross-section profiles different than the Gaussian one.
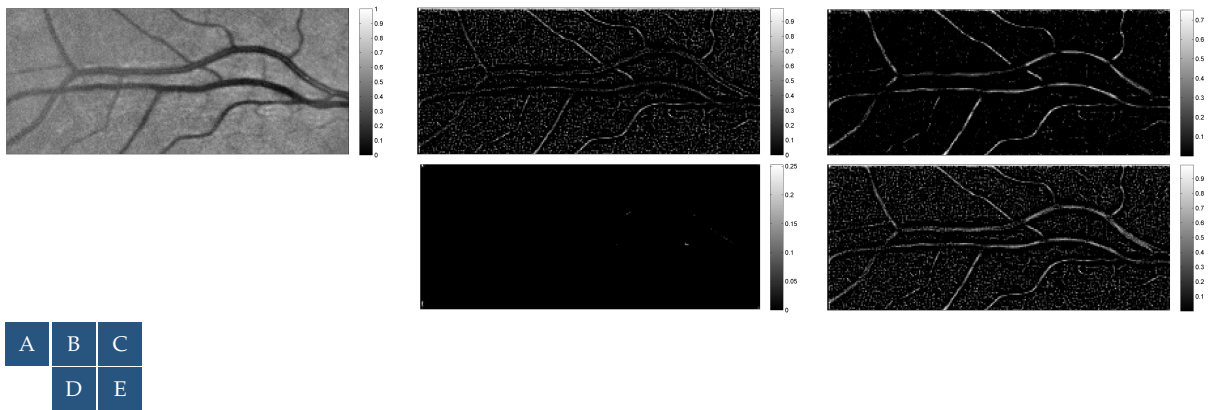
FIGURE 4.3: Fuzzy hit-or-miss curvilinear object detector employing structuring elements with different size. (A) Vessels in a 104×262-pixel image, processed with {3, 5} in (B); {5, 9} in (C); {9, 13} in (D); and {3, 5, 9, 13} in (E). The rest of parameters remain unchanged: structuring elements at 6 orientations, $C = T_\mathbf{L}$, $I = I_\mathbf{L}$ and $\text{Agg}_1 = \text{Agg}_2 = S_\mathbf{P}$.

Second, the number of orientations is also customizable, which may be useful to obtain finer-grained results at the cost of higher computational requirements. This is shown in Fig. 4.4, in which structuring elements with more orientations provide more curvilinear objects in this finger-print image, that includes curved ridges forming a circular shape. For m different orientations, we compute structuring elements as $\alpha_k = \frac{180°}{m}k$ for $k = 1, \ldots, m$.

Third, an interesting parameter to customize is the aggregation across different scales and orientations. We think of Agg_1 and Agg_2 as functions with a disjunctive-like behaviour, since we want to *add* the results along several orientations and sizes. Among other possibilities, we employed the maximum $S_\mathbf{M}$, the probabilistic sum $S_\mathbf{P}$ and the Łukasiewicz t-conorm $S_\mathbf{L}$. We also used the arithmetic mean in order to support the claim that Agg_1 and Agg_2 must have a disjunctive-like behaviour. We remark that the arithmetic mean presents an averaging-like behaviour, whereas the other operators are t-conorms. Besides, since t-conorms are associative, they can be extended to aggregate n variables in a straightforward manner. From an intuitive point of view, one very high output (both across orientations and scales) seems more important than several medium-level outputs in order to detect a curvilinear object, since a strongly pronounced curvilinear object in a specific direction should be favoured over averagely pronounced objects in several directions. In our experiments, however, we found that the best results were obtained by aggregations with an
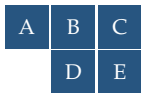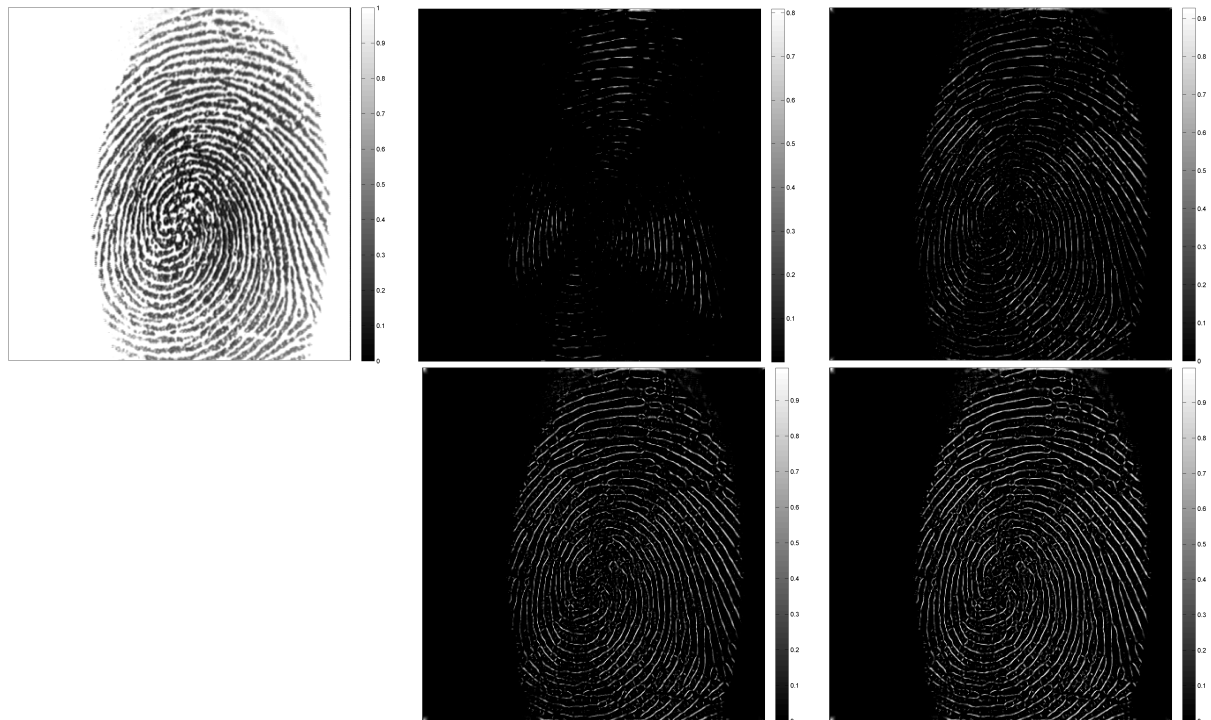
FIGURE 4.4: Different results of the detector for the 374×388-pixel fingerprint image (A), with 2 orientations (B), 4 orientations (C), 6 orientations (D) and 8 orientations (E). The rest of parameters remain unchanged: structuring elements of sizes $s \in \{5, 9, 13\}$, $C = T_{\mathbf{L}}$, $I = I_{\mathbf{L}}$ and $\mathrm{Agg}_1 = \mathrm{Agg}_2 = S_{\mathbf{P}}$.

additive behaviour (*e.g.* the probabilistic sum or the Łukasiewicz t-conorm). This is explained by the limited resolution in both orientations and scales. For instance, since only a number of orientations can be tested, any curvilinear object may be partially captured by more than one orientation, and thus a maximum operator would underestimate its real impact. As can be seen in Fig. 4.5, we found that such additive behaviours led to an increased contrast of small curvilinear objects in the final estimation. We decided to set the probabilistic sum as default. It provides good visual results and is preferrable to the Łukasiewicz t-conorm due to its lower saturation when the number of values to aggregate increases.

Lastly, the fuzzy operators C, I, N used are of paramount importance, since they control the ultimate behaviour of the algorithm. This affects the degree with which the structuring element is required to fit into the image, and the degree with which a brighter or darker background affects the final estimation. In this work, we leverage Proposition 4.1 to avoid capturing flat regions or areas with low-level noise and, thus, enhance even more the curvilinear structures. The hypotheses that such proposition requires are fulfilled when processing (i) a flat region with (ii) B_1 reaching both 0 and 1, (iii) using, for instance, the Łukasiewicz implication $I_{\mathbf{L}}$ and (iv) employing the Łukasiewicz t-norm $T_{\mathbf{L}}$. We remark that, when employing the Łukasiewicz implication $I_{\mathbf{L}}$, the fuzzy erosion becomes a shifted version of the grayscale erosion based on the umbra approach truncated in $[0, 1]$. Thus, it would be possible to design the same hit-or-miss operator using other techniques. However, we keep using the fuzzy erosion due to its interpretability (*e.g.* to measure *how much the image contains the structuring element*). Whenever any of these conditions is not fulfilled, the background leaves a residual value that partly masks the location of curvilinear objects. For a depiction of this effect, see Fig. 4.6.
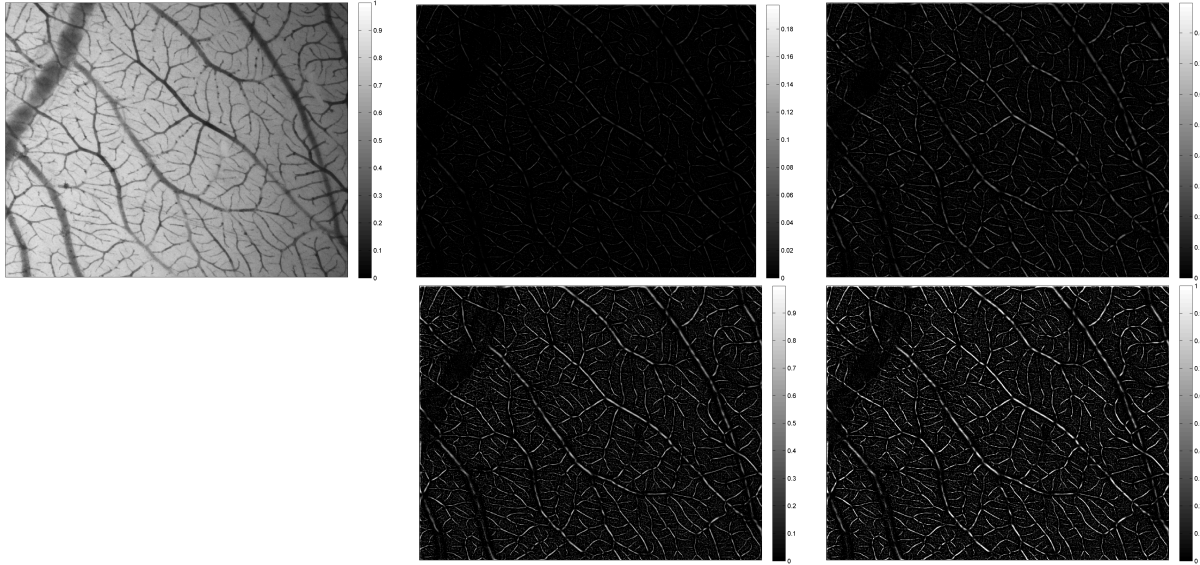
FIGURE 4.5: Aggregation of fuzzy hit-or-miss with different aggregation functions. 480×640-pixel image of tissue exhibiting angiogenesis (A). For both Agg_1 and Agg_2, we employed the mean (B), the maximum (C), the probabilistic sum (D) and the Łukasiewicz t-conorm (E). The rest of parameters remain unchanged: structuring elements of sizes $s \in \{5, 9, 13\}$ and 6 orientations, $C = T_{\mathbf{L}}$ and $I = I_{\mathbf{L}}$.
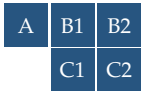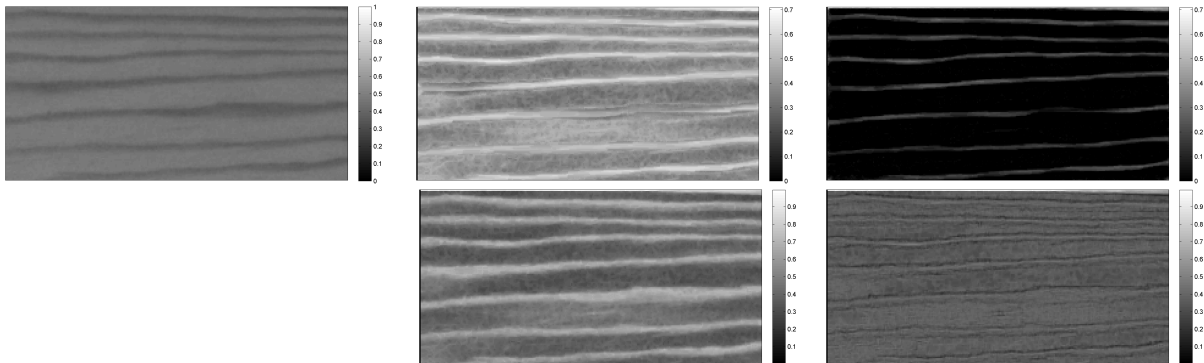
FIGURE 4.6: Fuzzy hit-or-miss curvilinear object detector with different structuring elements. Rings in a 132×280-pixel image of a wood plank (A), with normalized structuring elements, (B); and non-normalized ones, (C); employing the minimum t-norm, (1); and the Łukasiewicz t-norm (2). The aggregation function used is the maximum to improve the visualization of this effect. The rest of parameters remain unchanged: structuring elements of sizes $s \in \{5, 9, 13\}$ and 6 orientations, $I = I_{\mathbf{L}}$ and $Agg_1 = Agg_2 = S_{\mathbf{M}}$.

We emphasize again that this method only detects bright curvilinear objects. In our experiments, we applied the detector to the inverted grayscale image (*i.e.* $A'(x, y) = 1 - A(x, y)$) whenever we aimed at dark curvilinear objects. We also observe that this method was inspired by Bai *et al.*'s work [14]. In contrast with it, we employed grayscale structuring elements and non-crisp operators to produce a grayscale output instead of a binary one. A comparison between them is found in Sect. 4.4.

4.3 Curvilinear Object Detector based on the Fuzzy Black Top-Hat

The second curvilinear object detector we introduce is based on another fuzzy morphological operator, the fuzzy black top-hat operator. As already commented in Sect. 2.4, such operator enhances the regions that can not be recovered by an erosion after a dilation. Since dark curvilinear objects are relatively thin, they tend to completely disappear after a dilation. The erosion can not recover such regions and, therefore, the fuzzy black top-hat operator captures them.

In contrast with the detector based on the fuzzy hit-or-miss operator, this operator extracts curvilinear objects regardless of their orientation. Besides, there is no need to employ a multiscale approach. One single fuzzy black top-hat operator, when using a circular-shaped structuring element of size $s \times s$, enhances all objects thinner or equal than $(s-1) \times (s-1)$. Such a rounded shape is appropriate due to the behaviour of the top-hat operator: it extracts the regions that are completely erased with a fuzzy morphological dilation, and so we erase them isotropically. Thus, to select all curvilinear objects within a certain width range, $s_1 \leqslant w < s_2$ we need only two fuzzy black top-hat operations, as depicted in Fig. 4.7. Since the detector is designed to extract dark curvilinear objects, images that contain light curvilinear objects are inverted (*i.e.* $A'(x, y) = 1 - A(x, y)$) before being processed with the detector $\mathcal{D}_2(A)$.
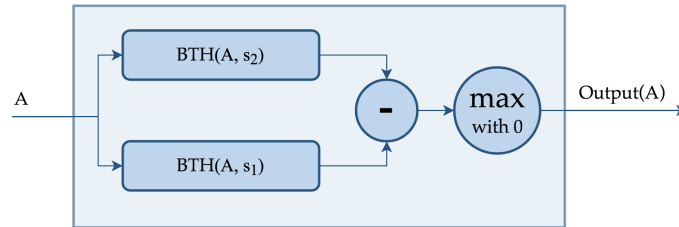
FIGURE 4.7: Workflow of the top-hat curvilinear object detector for curvilinear objects of width $s_1 \leqslant w < s_2$, where the subtraction step never outputs negative values, but a zero value instead. We remark that $\mathcal{BTH}(A, s_i)$ represents, in short, $\mathcal{BTH}_{C,I}(A, B^{(s_i)})$, where C, I should be specified and $B^{(s_i)}$ is a $s_i \times s_i$-pixel, Gaussian-shaped structuring element.

This detector also uses structuring elements as templates for the structures it aims to capture. In this case, two structuring elements as the ones shown in Fig. 4.8 are needed to segment curvilinear objects within a range.
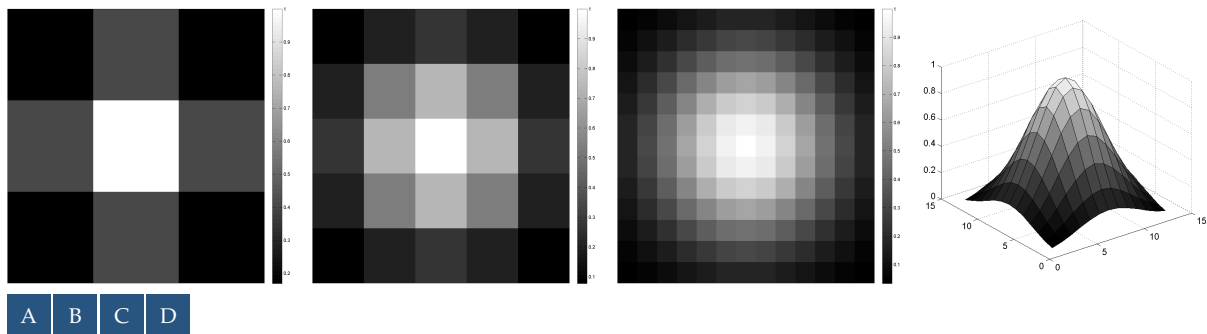
FIGURE 4.8: Structuring elements for the fuzzy black top-hat curvilinear detector, with size 3×3 (A), 5×5 (B), 13×13 (C) and a 3D view of its Gaussian decay of the 13×13 (D).

The curvilinear detector based on the fuzzy black top-hat operator is now formally specified. Let C and I be, respectively, a fuzzy conjunction, and a fuzzy implication function. Let A be a grayscale image. Then, the *curvilinear object detector based on the fuzzy black top-hat operator*, $\mathcal{D}_2$, is defined as:

$$\mathcal{D}_2(A) = \max(\mathcal{BTH}_{C,I}(A, B^{(s_2)}) - \mathcal{BTH}_{C,I}(A, B^{(s_1)}), 0) \qquad (4.2)$$

where $s_1, s_2 \in \mathbb{R}$ denote the length in pixels of the side of the structuring element; $B^{(s)}$ is a rounded $s \times s$-pixel structuring element with a Gaussian decay, having $\sigma = s/4$ and a maximum value of 1. The Gaussian decayment profile, similarly to the fuzzy hit-or-miss method, is based on the hypothesis that curvilinear objects have, approximately, a Gaussian cross-section profile (see Table 3.2, page 22). We remark that such value for the standard deviation has been manually chosen so that the bell curve shape covers the entire structuring element, as can be observed in Fig. 4.8. We also observe that negative values are set to zero. This is uncommon and does not affect regions that appear as curvilinear objects, but the monotonicity of the fuzzy top-hat operators is not guaranteed with respect to the used structuring element.

This detector can also be customized, both in terms of the fuzzy operators employed and on the sizes s_1 and s_2 of the structuring elements. Firstly, from the myriad of fuzzy operators that can be defined, we focus on the minimum t-norm $T_\mathbf{M}$, the product t-norm $T_\mathbf{P}$ and the Łukasiewicz t-norm $T_\mathbf{L}$ due to their extensive use and the properties they fulfill (see Table 2.1, page 11). As we observe in Fig. 4.9, the minimum t-norm is associated with stronger responses, yielding the Łukasiewicz t-norm the weakest ones. In all the experiments, we use the R-implication of the corresponding t-norm as the fuzzy implication function in order to guarantee that they do not yield negative values. Although the method is robust with respect to the fuzzy operators used, the product t-norm appears to be the one that provides the best contrast between the curvilinear objects and the rest of the image.

A B C D
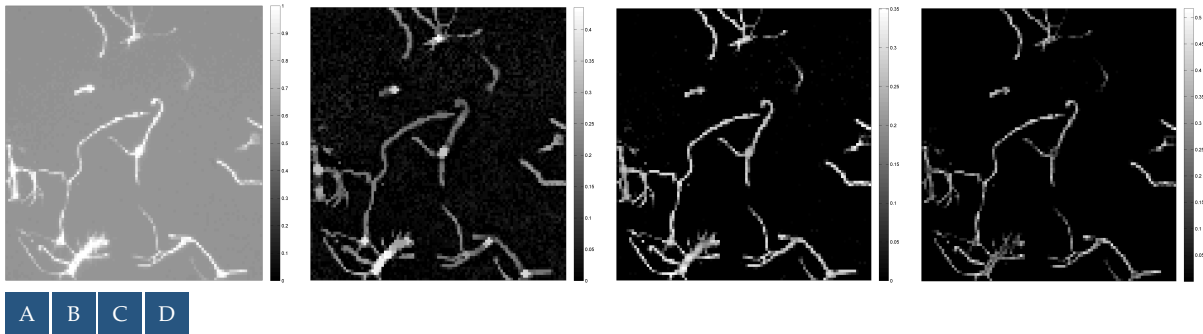
FIGURE 4.9: 50×50-pixel image depicting axons (A), and the fuzzy black top-hat curvilinear detector, using the minimum t-norm (B), the product t-norm (C) and the Łukasiewicz t-norm (D). The structuring elements remain unchanged, having sizes $s_1 = 3, s_2 = 13$.

Secondly, the size of the structuring elements can be customized to detect exactly the desired objects, since it captures curvilinear structures whose width w fulfills $s_1 \leqslant w < s_2$. Therefore, a single computation detects a whole range of curvilinear objects. As can be observed in Fig. 4.10, the objects detected with different structuring elements vary according to their size.

A B C D
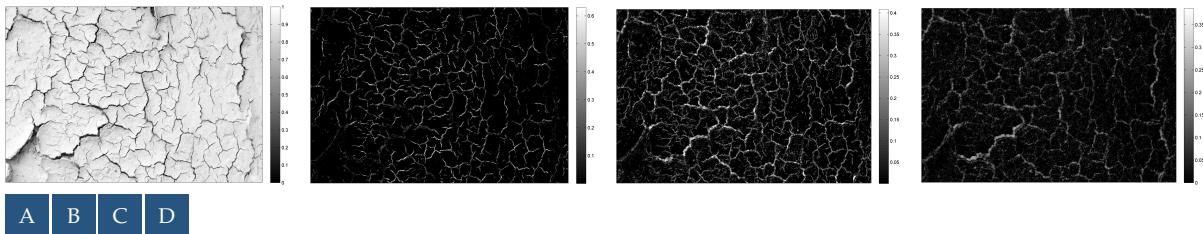
FIGURE 4.10: 350×550-pixel image of cracks in pavement (A), and the fuzzy black top-hat curvilinear detector, using $(s_1, s_2) = (1, 5)$ in (B), $(s_1, s_2) = (5, 9)$ in (C) and $(s_1, s_2) = (9, 13)$ in (D). The rest of parameters remain unchanged: $C = T_\mathbf{P}$ and $I = I_\mathbf{GG}$.

Unless remarked, the parameters used are the following ones: the conjunction C is the product t-norm $T_\mathbf{P}$, the implication I is its residual implication, the Goguen implication $I_\mathbf{GG}$, and the structuring elements have size 3×3 and 13×13.

In this work, we use fuzzy mathematical morphology, but other grayscale morphologies have been developed. One of them is based on the umbra approach [164]. In it, conjunctions and implications do not appear, and are replaced by the addition and subtraction of two numbers. Although they do not provide results in the interval $[0, 1]$, we can compute a top-hat operator from them. In Fig. 4.11 we depict a comparison between the two approaches to study their differences. As we can observe, their behaviour is quite similar. The fuzzy black top-hat enhances better the objects whose gray levels are closer to those of its background (*e.g.* the spirulina found on the upper right corner), and it detects better objects within a wider range of widths. On the other hand, it seems to introduce slightly more noise.

| A1 | A2 | A3 |
| B1 | B2 | B3 |
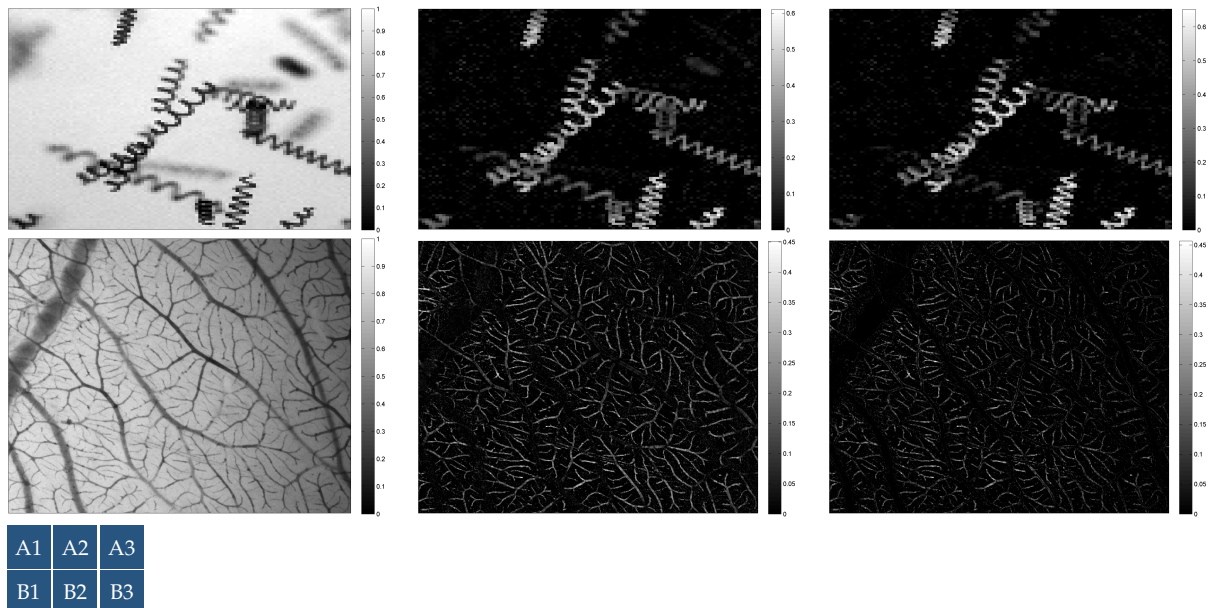
FIGURE 4.11: (1) Original and curvilinear object detectors based on (2) the fuzzy black top-hat and on (3) the umbra-approach top-hat of a detail of (A) spirulina algae through microscope and vessels in (B) a chicken chorioallantoic membrane. Both top-hat transforms use structuring elements of sizes $s_1 = 3, s_2 = 13$, and the fuzzy top-hat transform also employs $C = T_\mathbf{P}$ and $I = I_{\mathbf{GG}}$.

Lastly, this method presents two shortcomings that must be acknowledged. First, similarly to the detector presented in Sect. 4.2, it only detects objects that are darker than their surroundings. If the objects of interest are brighter than their background, the detector could employ the fuzzy white top-hat operation $\mathcal{WTH}$ from Def. 2.24 (page 13), or we must invert the image before processing it with the detector. However, to detect *both* dark and bright objects, we must process the image twice, doubling the amount of resources needed. Second, this detector enhances thin objects, but does not distinguish between elongated objects and isolated ones. Although this does not pose a problem in the majority of situations, the output could be affected whenever very high levels of noise create shapes larger than the first structuring element.

4.4 Comparison of General-Purpose Curvilinear Object Detectors

In the following, we compare the performance of the two algorithms presented in Sects. 4.2 and 4.3 with other state-of-the-art algorithms. Specifically, we employ four images to provide a visual comparison. Additionally, a quantitative comparison between these algorithms is found in Section 5.2. In it, the methods are employed to segment vessels from the retinal images of the DRIVE dataset [162].

4.4.1 Methodology

In this section we provide visual comparisons of general-purpose curvilinear detectors. We do not include quantitative performance measures since we are not aiming at solving a specific problem: we are comparing generic curvilinear detectors that may be used in different situations, exhibiting thus a different behaviour. Since we are dealing with methods that can be customized to specific situations, it would be unfair to compare them as long as they are not adapted to the needs of each specific application. Instead, we rely on the visual comparison of the results when processing images with different features to let the practitioner choose the approach that best fits their needs.

We compare the two fuzzy morphological detectors presented in the paper, the curvilinear detector by Bai *et al.* [14], the grayscale hit-or-miss method by Barat [16], the curvilinear detector by Krylov and Nelson [99], the path openings operator [168] and the RORPO algorithm [128]. These last two methods are computed with the source code provided by their authors, whereas the others have been implemented in MATLAB® following the steps described in the corresponding article. In addition, a ground truth has also been manually delineated to provide a binary segmentation. We remark the singularity of this task: edge detectors are not appropriate since we are not concerned with the limits between two objects, but with a tubular-shaped, elongated object located on a uniform or non-uniform background.

We remark that different methods provide results in different formats. Our morphological detectors provide a grayscale output that can be interpreted as the degree of belonging of each specific pixel to a curvilinear object. The method by Bai *et al.*, based on the morphological skeleton, provides a binary mask of the centerlines of the curvilinear objects found [14]. Krylov and Nelson's method, on the other hand, extracts line segments and represents them in terms of their position, orientation and strength due, again, to its nature, which relies on the Radon transform. The path openings method, originally introduced by Heijmans *et al.* [81], provides a transformed image where the curvilinear objects have been enhanced. For computing path openings we used the algorithm proposed by Talbot and Appleton in [168]. The RORPO operator by Merveille *et al.* [128, 127] is based on path openings and outputs a similar image, where structures with one clear direction are favoured. Since path openings and RORPO are designed to extract white curvilinear objects, we inverted the image (*i.e.* $A'(x) = 1 - A(x)$) when the curvilinear objects to be detected were darker than their background.

In addition to these published methods, we also compare our methods with a grayscale hit-or-miss operator. Such a grayscale hit-or-miss operator, proposed by Barat [16], is an example of grayscale hit-or-miss based on the umbra approach [164]. It has been used to implement a curvilinear detector as in Sect. 4.2, in order to compare the fuzzy and grayscale version of the hit-or-miss. We emphasize that a comparison with a grayscale operation is reflected in Fig. 4.11.

The methods are computed with the following parameters. Path openings are computed with $L = 60, K = 2$; the RORPO operator with $L_{min} = 20$, a single scale, and robustness parameter $R = 1$; the fuzzy hit-or-miss curvilinear object detector is executed along 6 orientations (*i.e.* $\alpha_k = k \cdot 30°$ for $k = 1, \ldots, 6$), with structuring elements of size $s_1 = 5, s_2 = 9, s_3 = 13$, the Łukasiewicz t-norm T_L, the Łukasiewicz implication I_L, the standard negation and the probabilistic sum S_P as both aggregation functions; lastly, the fuzzy black top-hat results correspond to the product t-norm T_P, its residual implication I_{GG} and structuring elements of size $s_1 = 3, s_2 = 13$. Besides, the Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm is used as a preprocessing step in all images (see 2.5.2, page 14). It is an contrast enhancement technique that re-maps the grayscale values of an image so that they are more equidistributed, providing good results in a variety of settings, and in particular in medical imagery, for which it was originally designed. The CLAHE has been applied with a clip limit of 0.01 and an 8×8 subgrid to provide the adaptive behaviour. No postprocessing step is applied in order to better visualize the behaviour of the detectors.

4.4.2 Visual Results

The first comparison image, Fig. 4.12, is a clear fingerprint with a resolution of 349×545 pixels. Being the easiest image in this comparison, all methods except for Krylov and Nelson's perform reasonably well. Given the nature of the task (*i.e.* detecting the frequency or structure of ridges, among others [197]), the best methods seem to be Bai *et al.*'s one and the fuzzy hit-or-miss based method. Although the fuzzy black top-hat detector and the RORPO operator provide similar outputs, the latter one favours much more the curvilinear shapes (see, for instance, the blocking effect due to the JPG compression). The grayscale hit-or-miss by Barat [16] provides low-contrast curvilinear objects, besides detecting the borders rather than the curvilinear objects themselves. The path opening of the original image results in an enhanced version: it lefts the image almost unaltered.
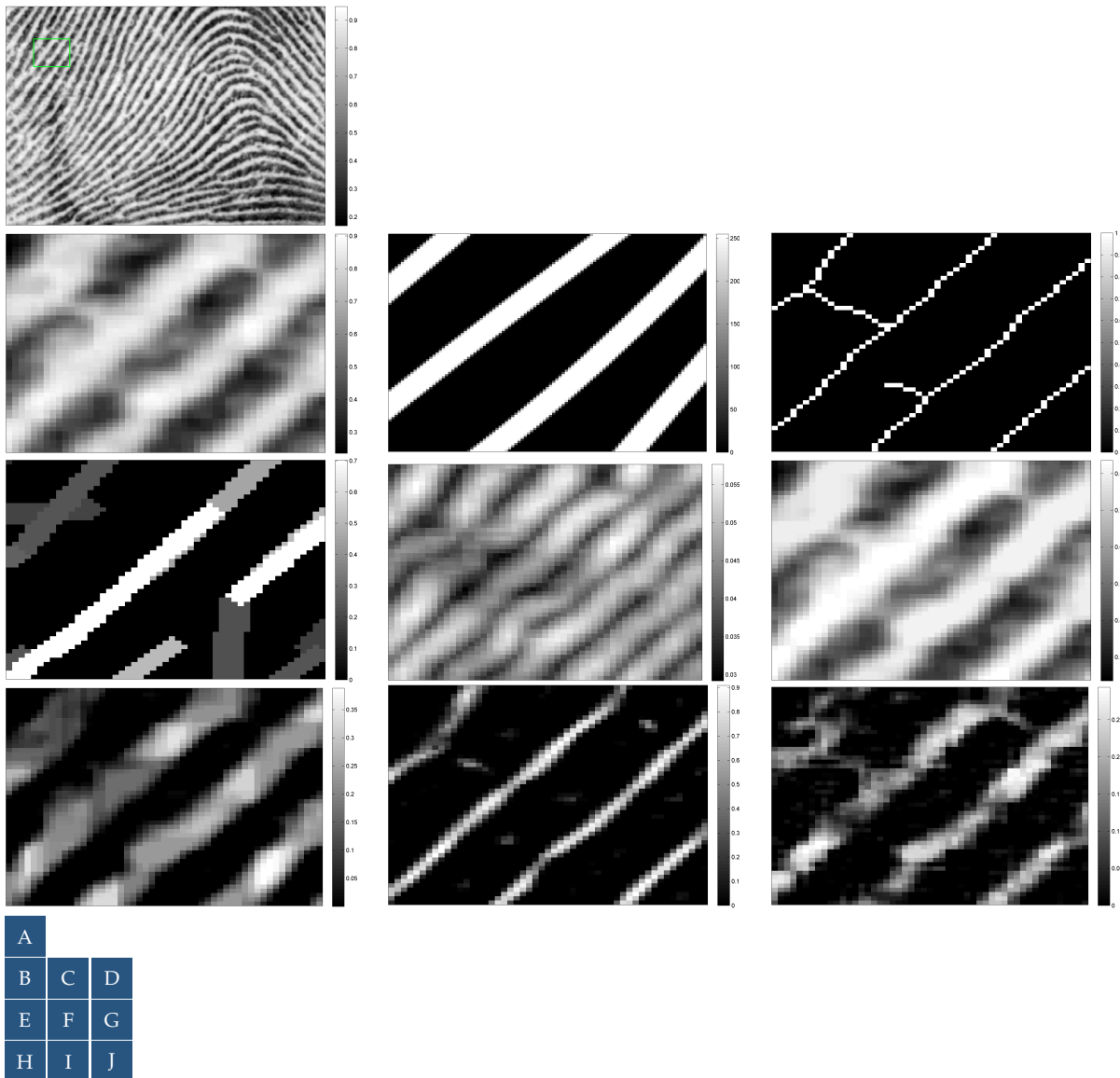
FIGURE 4.12: Detail of a fingerprint (50×50 pixels). Path openings are computed with $L = 60$, $K = 2$; the RORPO operator with $L_{\min} = 20$, a single scale, and robustness parameter $R = 1$; the hit-or-miss methods employ structuring elements of sizes $s \in \{5, 9, 13\}$ and 6 orientations, and the fuzzy one $C = T_{\mathbf{L}}$, $I = I_{\mathbf{L}}$ and $\mathrm{Agg}_1 = \mathrm{Agg}_2 = S_{\mathbf{P}}$; lastly, the fuzzy black top-hat method employs structuring elements of sizes $s_1 = 3, s_2 = 13$, $C = T_{\mathbf{P}}$ and $I = I_{\mathbf{GG}}$. (B) Detail of original image. (C) Ground Truth. (D) Bai *et al.* [14] (90.84 s). (E) Krylov and Nelson [99] (38.43 s). (F) Grayscale hit-or-miss [16] (47.48 s). (G) Path Openings [168] (4.17 s). (H) RORPO [127] (57.64 s). (I) Fuzzy hit-or-miss method (52.12 s). (J) Fuzzy black top-hat method (13.46 s).

Fig. 4.13 is a 339×446-pixel medical image of a common carotid artery obtained with B-mode

ultrasounds. A piece of valuable subjacent information in these images is the thickness of the intima-media (*i.e.* the inner layer of the artery's wall) [114]. To tackle this problem, the fuzzy black top-hat based method seems adaptable enough to detect the walls of the artery and to measure its maximum width. The RORPO operator successfully detects strong curvilinear objects, but does not detect small curvilinear structures near strong ones. On the other hand, Bai *et al.*'s algorithm falls short to identify either the thickness of the wall or to clearly differentiate the wall from the noisy background, as does Krylov and Nelson's algorithm. Lastly, the grayscale hit-or-miss by Barat and the fuzzy hit-or-miss algorithm correctly detect the outer wall, but fall short to identify the inner wall of the artery. Similarly to the previous case, the path opening transform slightly softens the image but preserves its texture.
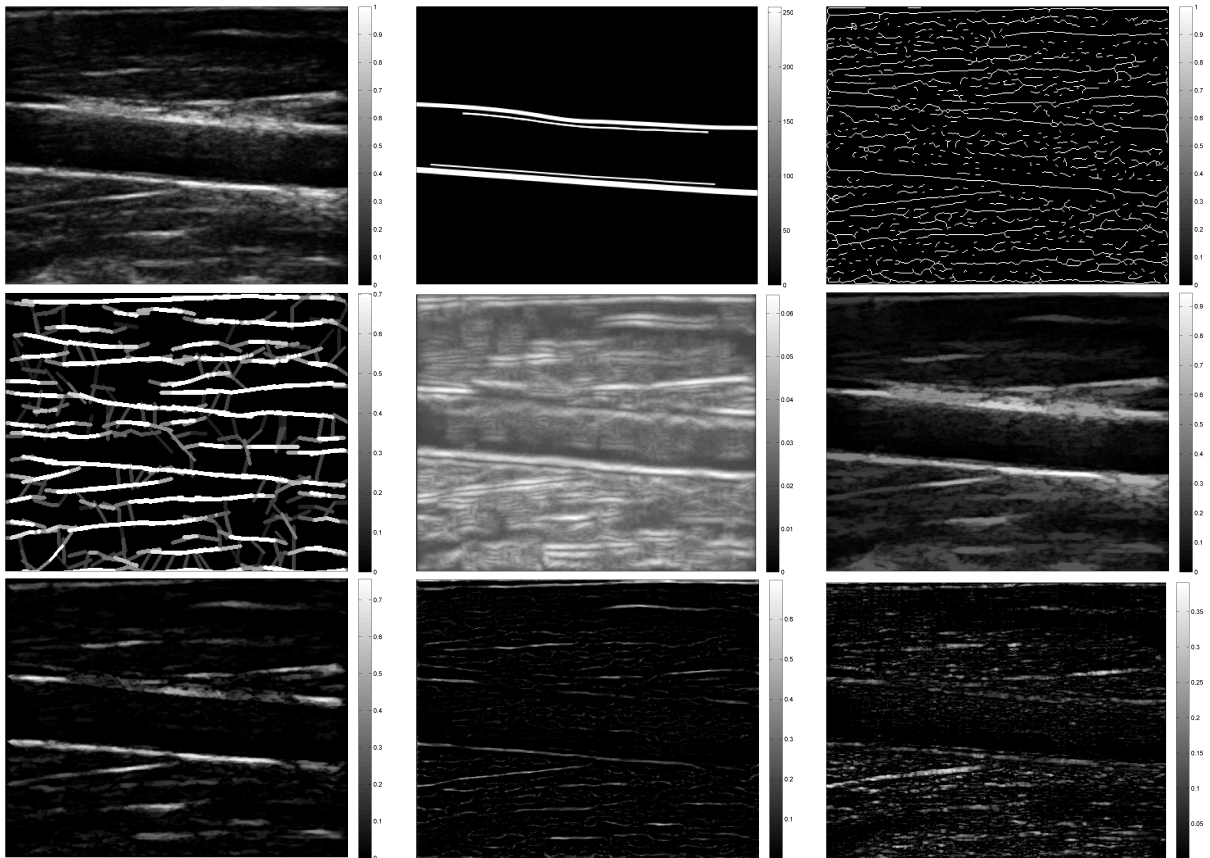
FIGURE 4.13: Common carotid artery in a B-mode ultrasound image (339×446 pixels). Path openings are computed with $L = 60$, $K = 2$; the RORPO operator with $L_{\min} = 20$, a single scale, and robustness parameter $R = 1$; the hit-or-miss methods employ structuring elements of sizes $s \in \{5, 9, 13\}$ and 6 orientations, and the fuzzy one $C = T_{\mathbf{L}}$, $I = I_{\mathbf{L}}$ and $\text{Agg}_1 = \text{Agg}_2 = S_{\mathbf{P}}$; lastly, the fuzzy black top-hat method employs structuring elements of sizes $s_1 = 3$, $s_2 = 13$, $C = T_{\mathbf{P}}$ and $I = I_{\mathbf{GG}}$. (A) Original image. (B) Ground Truth. (C) Bai *et al.* [14] (71.84 s). (D) Krylov and Nelson [99] (59.77 s). (E) Grayscale hit-or-miss [16] (50.18 s). (F) Path Openings [168] (2.75 s). (G) RORPO [127] (45.38 s). (H) Fuzzy hit-or-miss method (38.44 s). (I) Fuzzy black top-hat method (10.62 s).

The next sample is a 371×446-pixel detail of a grayscale picture containing human wrinkles. Human wrinkles detection has been applied to estimate the age of people or as a form of soft biometric identification [175, 54]. As far as our experiments go, the method whose visual inspection is better is the fuzzy hit-or-miss one, as can be observed in Fig. 4.14. We remark that the method by Bai *et al.* falls short when processing flat but dark areas, since the estimation in such areas do not

correspond to any curvilinear structures but to random noise. On the other hand, the grayscale hit-or-miss operator by Barat [16] seems to reflect the texture of the image rather than the curvilinear objects present in it. The fuzzy black top-hat based method presents a better contrast than the method by Bai *et al.* between the actual wrinkles and the noise in the top-right region, but it is lower than in the case of the fuzzy hit-or-miss method. The RORPO operator recovers wider regions, accounting for the total width of the wrinkles. It also detects some regions that contain noise, but outputs lower values than in the case of wrinkles. Again, Krylov and Nelson's method completely misses the extraction of curvilinear structures according to the goal of the task, which in this case is detecting the wrinkles. In this case, the path openings detector shows the opening of the inverse image, since we aim at detecting dark curvilinear objects. A lot of texture is preserved, and non-uniform illumination is responsible of the very large outputs in flat regions.
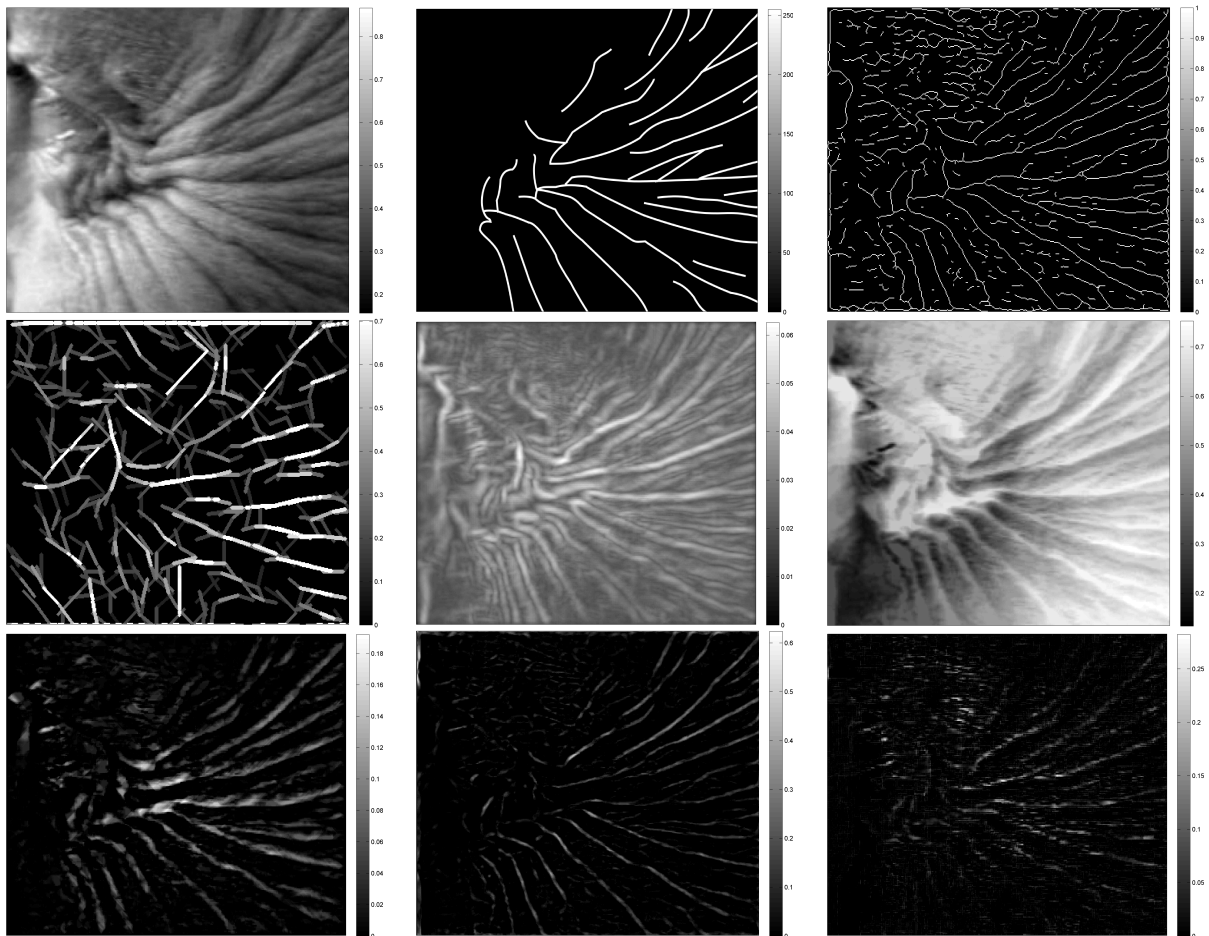
FIGURE 4.14: Wrinkles in a grayscale image of a human face, with an eye facing left (371×446 pixels). Path openings are computed with $L = 60$, $K = 2$; the RORPO operator with $L_{\min} = 20$, a single scale, and robustness parameter $R = 1$; the hit-or-miss methods employ structuring elements of sizes $s \in \{5, 9, 13\}$ and 6 orientations, and the fuzzy one $C = T_{\mathbf{L}}$, $I = I_{\mathbf{L}}$ and $\text{Agg}_1 = \text{Agg}_2 = S_{\mathbf{P}}$; lastly, the fuzzy black top-hat method employs structuring elements of sizes $s_1 = 3$, $s_2 = 13$, $C = T_{\mathbf{P}}$ and $I = I_{\mathbf{GG}}$. (A) Original image. (B) Ground Truth. (C) Bai *et al.* [14] (79.62 s). (D) Krylov and Nelson [99] (37.48 s). (E) Grayscale hit-or-miss [16] (51.37 s). (F) Path Openings [168] (2.95 s). (G) RORPO [127] (42.03 s). (H) Fuzzy hit-or-miss method (41.18 s). (I) Fuzzy black top-hat method (10.55 s).

Finally, the last application of curvilinear object detection with which we compare the algorithms is wire detection in low-altitude flight imagery. These images must be quickly processed

to detect high-voltage power lines that can pose a serious problem to aircrafts [152]. Despite their almost straight appearance, their low diameter makes wires potentially hard to detect, especially when projected on an irregular background. The fuzzy black top-hat method and the one by Bai *et al.* provide acceptable results, being the first one the only method capable of retaining information about the geometry of the objects from the original image. The grayscale hit-or-miss also provides good results, although each curvilinear object has been detected twice. In any case, both of them would require strong post-processing steps. The RORPO operator detects regions that are too wide, but also provides a strong output in the wires, and is not affected by the scattered noise that affects the fuzzy black top-hat method. While the fuzzy hit-or-miss method provides a weak output, Krylov and Nelson's method fails at capturing the segments of interest, which are too thin for them to be highlighted. In this last visual example, we can observe how path openings do not distinguish curvilinear objects from large uniform areas, such as the inner squares corresponding to the balconies.
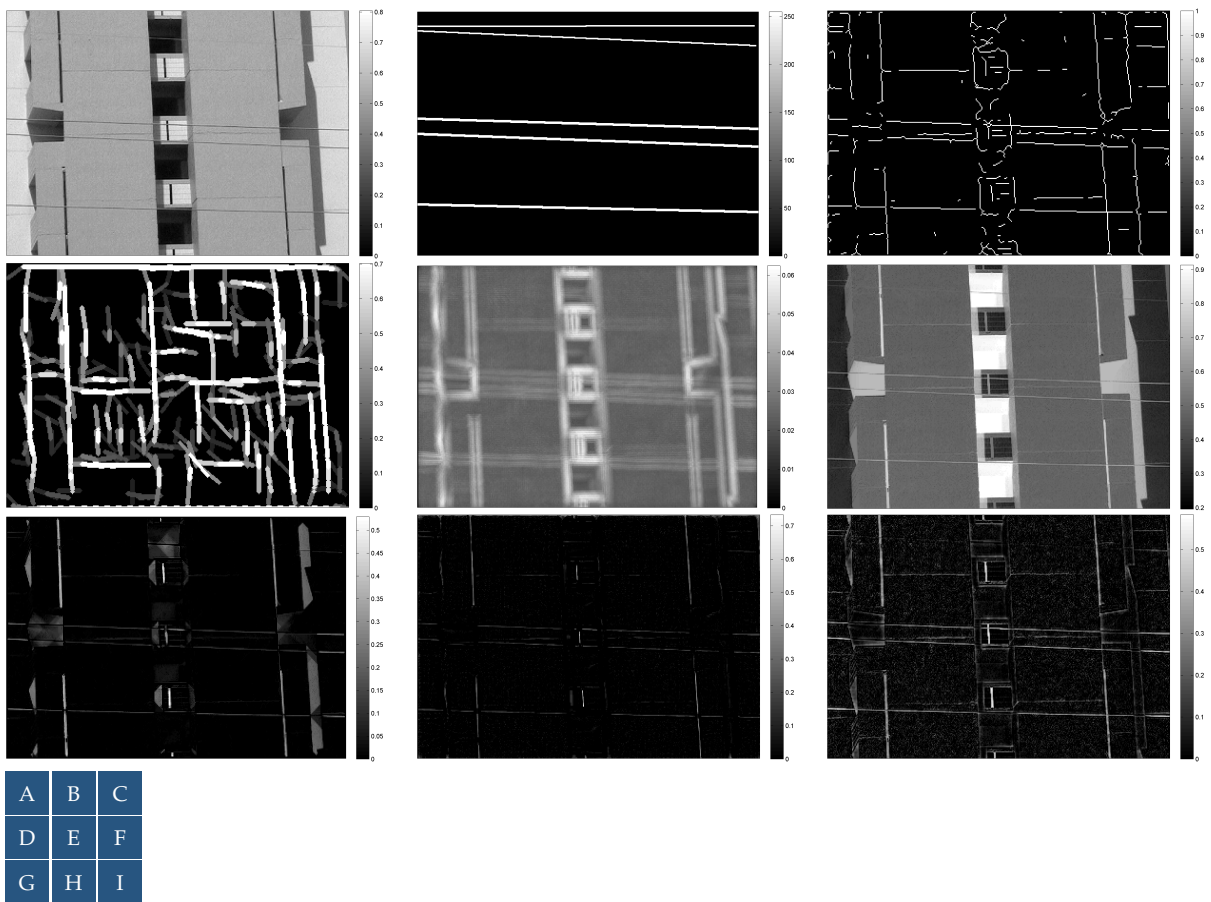
FIGURE 4.15: Power wires in urban environment (240×360 pixels). Path openings are computed with $L = 60$, $K = 2$; the RORPO operator with $L_{\min} = 20$, a single scale, and robustness parameter $R = 1$; the hit-or-miss methods employ structuring elements of sizes $s \in \{5, 9, 13\}$ and 6 orientations, and the fuzzy one $C = T_{\mathbf{L}}$, $I = I_{\mathbf{L}}$ and $\text{Agg}_1 = \text{Agg}_2 = S_{\mathbf{P}}$; lastly, the fuzzy black top-hat method employs structuring elements of sizes $s_1 = 3, s_2 = 13$, $C = T_{\mathbf{P}}$ and $I = I_{\mathbf{GG}}$. (A) Original image (B) Ground Truth (C) Bai *et al.* [14] (41.18 s) (D) Krylov and Nelson [99] (30.76 s) (E) Grayscale hit-or-miss [16] (29.08 s) (F) Path Openings [168] (1.42 s) (G) RORPO [127] (14.02 s) (H) Fuzzy hit-or-miss method (22.21 s) (I) Fuzzy black top-hat method (7.97 s)

4.4.3 Analysis of the Comparison

In this section, we compare the proposed methods from Sects. 4.2 and 4.3 and the rest of the state-of-the-art general-purpose algorithms based on the visual examples above. The method based on the fuzzy hit-or-miss operator outstands when extracting wide objects in their entirety and

produces highly contrasted estimations (see Fig. 4.12(D)). The method based on the fuzzy black top-hat operator also succeeds in extracting thin curvilinear objects (see the inner layer of the lower wall in Fig. 4.13(I)), which are clearly separated from the background even if they are close to bigger curvilinear structures. Besides, it detects bifurcations and crossovers more successfully than the hit-or-miss method (see, for instance, the details in Fig. 4.12). However, it does not suppress noise as well as the hit-or-miss based one, providing less-contrasted estimations. We emphasize that, although these methods are highly customizable, they are also robust with regard to their parameters. All images in the comparison section have been obtained with the exact same parameters, which gives an idea of the good overall performance of the algorithms, due to their well-grounded theoretical foundations within fuzzy mathematical morphology.

In contrast to them, the rest of methods do not provide a pixel-wise indicator of its degree of belonging to a curvilinear structure in the image. Bai *et al.*'s method outputs the centerlines of the curvilinear objects. Krylov and Nelson's method only provides the location, orientation and strength of the most significative segments. Path openings provide images where curvilinear objects are enhanced, but finer details are still present in the image. It suffers from non-uniform illumination problems when it is used as a curvilinear object detector. Besides, it does not discern between curvilinear objects and large blob-shaped objects. The same problems affect the grayscale hit-or-miss method with the grayscale operator by Barat [16]. The RORPO operator successfully discriminates curvilinear objects, and extracts them in their entirety (*i.e.* accounting for their full width). Its robustness to noise is achieved with a pre-processing stage in which the image is dilated, from which the path openings are computed. Such noise robustness, which is responsible of its good results, also induces a potential shortcoming: it does not discriminate well small curvilinear objects that are located near larger ones, in contrast to other methods such as the fuzzy black top-hat algorithm.

We remark that all code is executed in the same machine (Quad-core@3.10 GHz, 16 GB RAM) and using non-optimized MATLAB® code, except for the path openings algorithm, which uses the C implementation from Talbot and Appleton[1] [168]. Krylov and Nelson's method was also executed with their publicly available implementation[2]. The method by Merveille *et al.* was executed with their recent implementation, which they kindly sent us along with the corresponding paper [127].

4.5   Conclusion

In this chapter, we have presented two curvilinear object detectors based on the fuzzy mathematical morphology framework. They aim at capturing elongated, tubular-like objects—in contrast with edge detectors, that capture the limits between two different objects.

The objective of this chapter has been to provide easy-to-use curvilinear detectors to be used in a variety of situations. They are based on operators from the fuzzy mathematical morphology. These include, but are not limited to, vessel segmentation, road and bridge location in aerial imagery, crack detection in pavement and the ones presented in Sect. 4.4. This may prevent researchers working in different problems from re-discovering the same tools and techniques (see, for instance, the Dual-Gaussian model [110] and the Bi-Gaussian function [191], which specify the same mathematical model).

The presented detectors are thought to be a single step into a larger system. Therefore, the input can be firstly pre-processed to obtain uniform images, and the output can be post-processed to refine the estimation and enhance thus the objects of interest. With regard to the former, we employed the CLAHE algorithm [142] to enhance the contrast of images. When it comes to post-processing, the method by Bai *et al.* searches for the object centerlines with a morphological thinning

---

[1]Available at https://github.com/path-openings, 06.06.2018.
[2]Available at https://sites.google.com/site/vlkryl/iciar2013, 06.06.2018.

while Krylov and Nelson employed a clustering algorithm to extract the largest linear features by combining several short, straight segments. These strategies could also be adapted to postprocess the outcomes of the detectors introduced in Sects. 4.2 and 4.3.

The two presented detectors are compared with other state-of-the-art, general-purpose curvilinear object detectors. Figures 4.12-4.15 visually depict their similarities and differences. According to Sect. 4.4.3, our detectors are able to locate most curvilinear objects, and to supress both flat areas and noise. We purposely leave out of the comparison fine-tuned pre- and post-processing steps. They would have a deep impact on the curvilinear objects and would bias the comparison.

Finally, we highlight the grayscale nature of our detectors. Due to the use of fuzzy mathematical morphology operators and to their design, their output can be interpreted as how *similar* a structure is to a curvilinear object. This similarity is, of course, never crisp due to the existence of noise, limited resolution and even the subjective nature of the definition of a curvilinear object, as argued in Sect. 3.2.

Chapter 5

Retinal Vessel Segmentation

This chapter studies the problem of detecting vessels in retinal images, which represent an example of curvilinear objects. The detection of vessels is the first step towards an automatic diagnosis and in-depth study of retinal images to aid ophtalmologists. In particular, we design an algorithm using a generic curvilinear object segmentation algorithm. To find the most appropriate one, we consider all the curvilinear object detectors compared in Sect. 4.4. These results are compared among them and also with a manual segmentation.

Due to technological advances, high volumes of data are being acquired and should be quickly processed. Besides, the real-time nature of many applications, such as processing images while a patient is being examined, also requires fast yet accurate algorithms. Besides comparing curvilinear object detectors for the task of vessel segmentation, we propose a real-time algorithm. To do so, we employ one the fuzzy black top-hat detector, introduced in Sect. 4.3. It is based on the fuzzy mathematical morphology, which is able to handle the uncertainty present in this type of images (see Sect. 2.4). The real-time algorithm processes images from the DRIVE and STARE datasets, in average, in 37 ms and 57 ms respectively.

Sect. 5.1 introduces the problem of retinal vessel segmentation and the related literature in which it has been considered. The two curvilinear detectors introduced in Ch. 4 are employed in a more complex algorithm to segment vessels in retinal images in Sect. 5.2. Afterwards, we fine-tune the best approach to design a real-time algorithm, presented in Sect. 5.3. Compared to the state of the art, it presents an outstanding trade-off between performance and speed. Sect. 5.4 shows this by comparing it with several published algorithms, which are analysed in Sect. 5.5.

5.1 Introduction

Retinal eye-fundus images can be used by experts to detect eye diseases. They are pictures of the interior surface of the eye: the retina. Along with technological improvements and the decreasing cost of eye-fundus cameras, an automated algorithm to estimate the influence of such conditions can help to widespread their use. Being this the final goal, we focus on this chapter on an essential first step: the detection of vessels. The attributes of the tree-like vessel structure, such as their tortuosity or width, provide information towards correctly diagnosing illnesses. In particular, retinopathy of prematurity affects the morphology of the vascular tree; diabetic retinopathy may be assessed by detecting microaneurysms, easier to detect after excluding the vascular tree; the vascular tree can also be used to localize the optic disc, which reflects glaucoma; and other pathologies such as hypertension, stroke and arteriosclerosis can also be reflected on the retinal vasculature. We emphasize that these are some of the features that curvilinear objects present, as studied in Sect. 3.2.

Vessel segmentation in retinal images has been a largely studied topic during the last years. As introduced in Ch. 3, a number of techniques have been applied to vessel segmentation, including wavelets and Gabor filters, matched filtering, variational methods and mathematical morphology. Some of the methods that outstand are briefly introduced in the following for the sake of completeness. Hoover *et al.* [83] detect vessels with matched filtering techniques and a local thresholding,

besides introducing the STARE database (see Sect. 3.4 for a complete description). Chanwimaluang and Guoliang [48] presented a method that also uses matched filtering, to later apply a threshold based on local entropy. Niemeijer *et al.* [133] presented a method later improved by Staal *et al.* [162]: they compute a number of features per pixel and, to decide if a pixel represents a vessel or not, they use a *k*-Nearest Neighbors classifier. Besides, they published a public database, DRIVE (again, see Sect. 3.4). Soares *et al.* [161] published a method that yields better segmentations than the previous ones. They used multiscale Gabor filters to extract pixel-wise features that were processed with different classifiers, such as the Least Mean Square Error (LMSE), the *k*-Nearest Neighbours (*k*NN) and the so-called Gaussian Mixture Model (GMM). Regarding mathematical morphology techniques, Zana and Klein [199] presented a method in which the sum of top-hat transformations is postprocessed by means of the Laplacian operator. Mendonça and Campilho [125] detect vessel centerlines with directional differential operators, followed by a region growing method. Afterwards, Azzopardi *et al.* [11] presented a powerful approach by using several filters based on Difference of Gaussians. They aggregate those filter responses by using a weighted harmonic mean. The weights are carefully crafted in order to maximize the response of the system to a bar-shaped pattern. Roychowdhury *et al.* [149] used an iterative procedure to segment the vessels. They perform a morphological top-hat transformation and reiterated hysteresis with different thresholds. The halt condition is based on estimators of the performance measures from the current and previous iterations. Maji *et al.* [117] used 12 convolutional neural networks whose results were aggregated afterwards. Despite using specialized hardware (specifically, a Tesla® K-20C GPU), their model spent 55 seconds in average to estimate a single image. Although several methods have been published, all of them tend to fail when detecting thin vessels. In the rest of the chapter, we will study which curvilinear objects are more appropriate to do so, paying special attention to the uncertainty caused due to the noise that typically affects eye-fundus images.

Few methods have been introduced taking into account not only their performance but also their computational requirements. Alonso-Montes *et al.* [5] presented a method based on snakes that lasted an average of 230 ms to process a single medium-sized image. They obtain a rough estimation of the vessels by filtering, thresholding and shrinking by means of a morphological erosion. Such estimation, along with the edges of the image, is processed to fed the snakes that provide the final estimation. Bankhead *et al.* [15] introduced a simple and fast algorithm that needed just $66 - 81$ ms to segment the vessels in similar images. They use a specific wavelet transform, and obtain a vesselness measure by adding the responses of the better contrasted wavelet levels. The vesselness measure is thresholded so that 20% of the pixels are retained in the binary image. Afterwards, Argüello *et al.* [9] extended the work by Alonso-Montes. They improved the algorithm by adapting it to be efficiently executed in GPUs. Although their CPU implementation needs around 500 ms to process a single image, high speedups are achieved when being executed in a GPU..

5.1.1 Eye-Fundus Images

Firstly, we present a description of a typical eye-fundus image, shown in Fig. 5.1, focusing our attention on the features that characterize vessels as such. In order to construct a method that segments vessels we identify which features characterize a pixel as part of a vessel. Similarly to Sect. 3.2.1, we include both geometric and photometric features—so we can infer information regarding the shape of objects and their colour components. We emphasize that this description is an extended version of the one provided in Ch. 3. Likewise, it is important to describe the noise found in these images so it can be properly faced.

In this chapter we use eye-fundus images from two publicly available databases: DRIVE [162] and STARE [83], both already introduced in Sect. 3.4. They were acquired operating in color photography, providing RGB images, without contrast dye. DRIVE is a retinal image database that contains 20 training and 20 test images of mostly healthy patients, all of them compressed with JPG. The vessels in each test image, of size 565×584 pixels, have been manually segmented by two
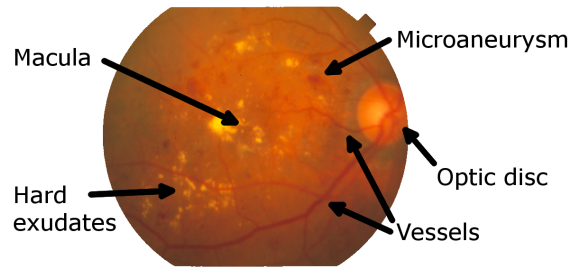
FIGURE 5.1: Annotated regions in the 14th image of the STARE database. The macula is hardly visible, whereas the illumination is noticeably non-uniform.

experts. On the other hand, STARE contains 20 images manually segmented and a number of retinal images without ground truth, stored in the lossless format PPM. Their size is 605×700 pixels and they contain more pathologies and artifacts than the images from the DRIVE database.

From a geometric point of view, vessels are connected regions that are thin, elongated 2D curvilinear objects that form a tree-like structure, with common bifurcations and few intersections. In the images considered, the width of vessels lies within a well-defined range, from sub-pixel resolution to up to 7-8 pixels. Besides, their width tends to be monotonically decreasing along their path. They start at the optic disc and grow towards the rest of the retina, surrounding the macula. The maximum curvature can be assumed to be bounded. Apperance of close vessels and tangent vessels occur frequently.

From a photometric point of view, vessels are always darker than their surrounding area, and their intensity changes gradually. The cross-section profile that they present is not clear among different researchers. It has been modelled as a Gaussian [199], a Difference of Offset Gaussians [125], a flat bar [11], and a flat bar convolved with a Gaussian [98]. A characteristic trait is the central vessel reflex, which causes the centerlines to be brighter than the rest of the vessel. It is originated by the presence of oxygen in the blood, which makes it specially salient in the case of arteries. We use the green channel due to the high contrast with which vessels outstand over the non-uniform background, offering the red and blue channels less information [69].

A combination of noise and artifacts complicate the task of segmenting vessels, as can be seen in Fig. 5.1. Retinal images have medium to low contrast, although technological improvements are making larger and better-quality images available. This includes high resolution imagery [135] and retinal videos [134, 62]. Moreover, these images are unevenly illuminated and often present a non-flat background texture. Besides the aforementioned vessels, these images include several differentiated regions: the macula, a darker circular region corresponding to the high-resolution vision; the optic disc, a bright circle that corresponds to the termination of the optic nerve; microaneurysms and other types of hemorrhages, that are darker and reddish areas; hard exudates, which appear as yellow points caused by lipid residues; cotton wool spots, being yellow/gray regions produced by damage to nerve fibers; and several others. These artifacts present a variety of features, and they should not be confused with vessels.

5.2 Curvilinear Detectors for Retinal Vessel Segmentation

In this section, we employ the general-purpose detectors designed in Ch. 4 to detect vessels in retinal images. We also use general-purpose curvilinear detectors by other authors (see Sect. 4.4) for the same task in order to provide an objective measure of their performance for this specific task. The analysis of the behaviour of the different curvilinear objects is the basis of a real-time vessel segmentation algorithm, which is introduced in Sect. 5.3.

To compare general-purpose curvilinear objects, we design a methodology to segment vessels from retinal images that uses a *generic curvilinear detector*. This procedure, depicted in Fig. 5.2, works as follows. First, the RGB image is projected onto its second component, the green channel, which is processed with the CLAHE algorithm (see Sect. 2.5.2). Second, a grayscale image containing only curvilinear objects has been computed for each test sample with the general-purpose curvilinear detector. Third, the pixels in the outer region, corresponding to the black area in the retinal image, have been deleted from the estimations. This mask is computed with a simple procedure introduced in Sect. 5.3. Finally, a threshold $t \in [0,1]$ has been applied to obtain a binary image and, thus, to compute the performance measures. Such threshold has been obtained as the one that maximizes the average F_1-score (see Sect. 3.4.2) for all the training images *simultaneously*. Of course, the performance indicator is biased, since we compute a parameter of the method, the threshold, from the samples used to validate it. However, we can still compare different curvilinear detectors, since all of them suffer a similar bias. We also remark that the F_1-score has been selected to avoid class skewness problems: only approximately 15% of pixels represent vessels.
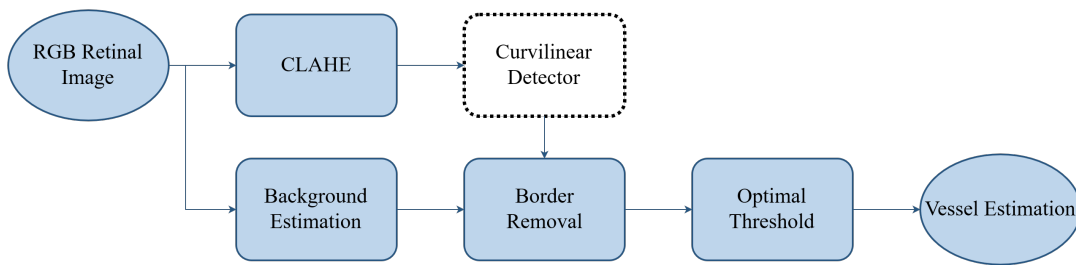

FIGURE 5.2: Flow diagram of the vessel segmentation algorithm for a general-purpose curvilinear detector.

In order to compare the curvilinear detectors based on their performance measures, we use all the training images in the DRIVE database [162]. The parameters of each method have been set based on the respective author's recommendations. In particular, path openings are computed with $L = 30$, $K = 5$; RORPO with $L_{\min} = 20$, two scales (1 and 1.5) and robustness parameter $R = 1$; the fuzzy hit-or-miss object detector with 6 orientations, $s \in \{5, 9, 13\}$, $C = T_{\mathbf{L}}$, $I = I_{\mathbf{L}}$ and $\mathrm{Agg}_1 = \mathrm{Agg}_2 = S_{\mathbf{P}}$; and the fuzzy top-hat object detector with $s_1 = 5$, $s_2 = 17$, $C = T_{\mathbf{P}}$ and $I = I_{\mathbf{GG}}$.

In Fig. 5.3 we can observe a comparison between different curvilinear detectors. In particular, for each method we show the grayscale output, in Figs. 5.3(1); and its binarization, in Figs. 5.3(2). It contains a sample with non-uniform illumination and that presents no pathologies. As we can observe, only the curvilinear detectors based on RORPO and on the fuzzy top-hat operator extract the whole tree of vessels. In particular, Bai *et al.*'s method and the fuzzy hit-or-miss object detector provide thinner objects.

The performance measures obtained by the method with each of the general-purpose curvilinear detectors are summarized in Table 5.1. In this comparison, we include the accuracy, the sensitivity, the specificity and the precision (see Sect. 3.4.2). To provide an insightful comparison of their computational complexity, we computed all of them in the same hardware. In particular, the system is an Intel® Core™ i5-3340 quad core @ 3.10 GHz with 16 GB of RAM, running Windows 10 Pro and MATLAB® R2017b.

5.2.1   Analysis of the Results

As we can observe in Table 5.1, the RORPO operator [127] and the fuzzy top-hat method outperform the rest of them. We emphasize that these quantitative results must be taken with caution: this is only one application where curvilinear objects must be segmented. Thus, the only valid conclusion is that the RORPO and the fuzzy top-hat operator are the curvilinear detectors better adapted to *this specific task*.

A1	A2	B1	B2
C1	C2	D1	D2
E1	E2	F1	F2
G1	G2	H1	H2
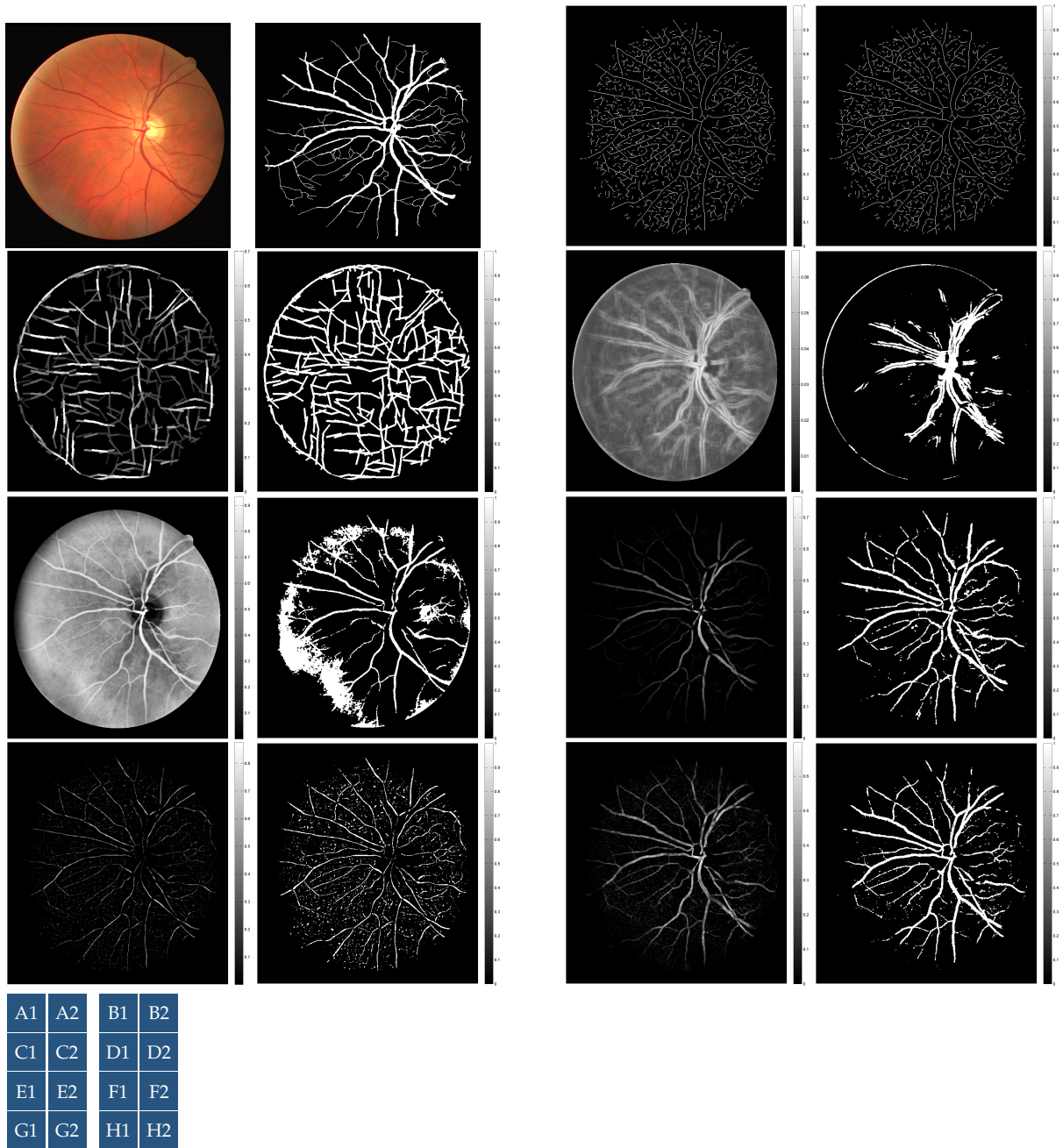
FIGURE 5.3: Comparison of results for the 4th sample of the DRIVE database: (1) curvilinear objects detected and (2) their binarization. (A1) original image and (A2) ground truth. (B) Bai *et al.* [14]. (C) Krylov and Nelson [99]. (D) Grayscale hit-or-miss [16]. (E) Path openings [168]. (F) RORPO [127]. (G) Fuzzy hit-or-miss object detector. (H) Fuzzy top-hat object detector.

Besides comparing their performance, Table 5.1 also provides information regarding their complexity. As we can observe, the fuzzy top-hat method is much simpler—and, thus, faster—than the rest of options. We emphasize that all of the methods have been executed in the same computer. The path openings [168] and the RORPO operator [127] have been executed with the code provided by the authors, whereas the rest of them are implemented using non-optimized MATLAB code.

This comparison only represents the performance of the curvilinear object detectors *as such*, without post-processing and without excessive parameter tuning. Methods specifically tailored to one specific curvilinear detector may provide better results, such as the method based on path openings by Sigurðsson *et al.* [160]. This is the motivation for the algorithm in Sect. 5.3, in which the fuzzy top-hat operator is employed to achieve a very competitive method both in terms of

TABLE 5.1: Results of methods with the 20 training samples of the DRIVE database [162], ordered by execution time. The results are given as the mean (and standard deviation) of the corresponding metric. They are compared against the 2nd ground truth results, in bold typography, using the Unilateral Wilcoxon Signed-Rank test with $\alpha = 0.05$ (see Sect. 3.4.1). The superscripts over the rest of results indicate whether $\sim$ the results are *not* statistically better/worse, $\dagger$ the results are statistically better, or $\ddagger$ the results are statistically worse. No superscript or missing values indicate that the individual segmentations were not available.

Method	Time per sample	Accuracy	Sensitivity	Specificity	F_1-score
2nd ground truth	—	**0.947** (0.005)	**0.773** (0.059)	**0.973** (0.008)	**0.786** (0.020)
Bai *et al.* [14]	168.18 s (0.360 s)	$0.877^{\ddagger}$ (0.013)	$0.208^{\ddagger}$ (0.017)	$0.975^{\sim}$ (0.005)	$0.301^{\ddagger}$ (0.021)
Krylov and Nelson [99]	29.088 s (3.272 s)	$0.669^{\ddagger}$ (0.013)	$0.252^{\ddagger}$ (0.017)	$0.729^{\ddagger}$ (0.012)	$0.162^{\ddagger}$ (0.015)
Grayscale hit-or-miss [16]	15.378 s (0.055 s)	$0.866^{\ddagger}$ (0.012)	$0.081^{\ddagger}$ (0.050)	$0.980^{\dagger}$ (0.007)	$0.130^{\ddagger}$ (0.073)
Path openings [168]	16.212 s (0.653 s)	$0.840^{\ddagger}$ (0.093)	$0.583^{\ddagger}$ (0.146)	$0.878^{\ddagger}$ (0.122)	$0.503^{\ddagger}$ (0.083)
RORPO [127]	20.465 s (0.448 s)	$0.934^{\ddagger}$ (0.007)	$0.686^{\ddagger}$ (0.061)	$0.970^{\sim}$ (0.010)	$0.724^{\ddagger}$ (0.030)
Fuzzy hit-or-miss	18.645 s (0.045 s)	$0.880^{\ddagger}$ (0.008)	$0.350^{\ddagger}$ (0.024)	$0.957^{\ddagger}$ (0.006)	$0.424^{\ddagger}$ (0.019)
Fuzzy top-hat	1.834 s (0.011 s)	$0.897^{\ddagger}$ (0.008)	$0.616^{\ddagger}$ (0.087)	$0.938^{\ddagger}$ (0.016)	$0.600^{\ddagger}$ (0.042)

performance and speed.

5.3   Real-time Retinal Vessel Segmentation

In Sect. 5.2 we studied the vessel segmentation problem with different general-purpose curvilinear detectors, and we found the fuzzy top-hat based curvilinear detector to be very appropriate for this task. Besides, the fuzzy top-hat is a very efficient operator in terms of its computational requirements.

The popularization of fundus cameras and retinal images and other technological advances provide an increasing amount information of the retina. These sources provide enormous amounts of data that should be processed quickly and efficiently. The manual processing of these amounts of data is unaffordable and, thus, automatic algorithms are required. Afterwards, this information can be further processed to extract indicators for some pathologies. For instance, after the vessels are identified, counting the red spots that characterize early diabetic retinopathy becomes an easier task [87]. Also, inpainting the vessels in the optic disc can ease the measurements of the neuroretinal rim to detect glaucoma [39].

This section is aimed at designing, implementing and optimizing a real-time yet accurate algorithm to segment vessels. Based on our previous results, the algorithm presented is developed taking into account both the utility of the operations used and their cost. We use operators from the fuzzy mathematical morphology to extract vessels taking into account their geometrical and photometric features introduced in Sect. 5.1. In particular, we use a fuzzy top-hat operator to enhance vessels. By reducing the computational cost of all the operations involved, we design an algorithm with high performance measure values but with a low processing time.

5.3.1 Description of the Method

Figure 5.4 depicts the steps carried out by our algorithm. It is a fine-tuned version of the digram for general-purpose curvilinear detectors in 5.2. Each step is designed to leverage one or more of the features described in the model for vessels. The stages of the algorithm are introduced as follows.
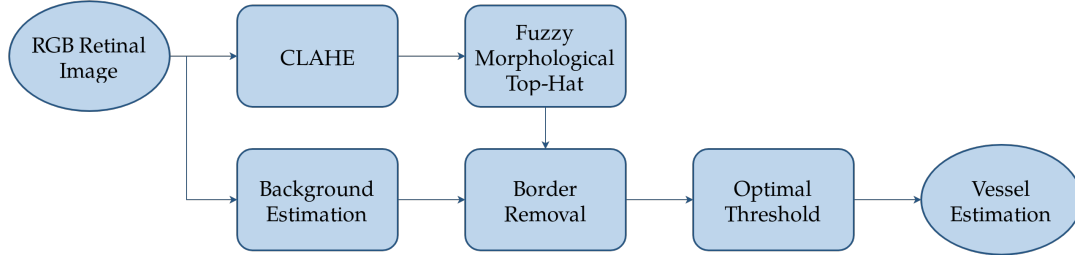

FIGURE 5.4: Flow diagram of the real-time vessel segmentation algorithm.

- **Preprocessing.** Firstly, the green channel of the RGB retinal image is selected. It is the one containing a better-contrasted image [161, 162], being the red one oversaturated and the blue one undersaturated.

  Secondly, similarly to the previous case, the CLAHE algorithm, is used as an enhancement step. In addition to further improving the contrast, it eases the uneven illumination problem without worsening the effect of the non-flat background texture. The clip limit is set to two and the subgrid in which we divide the image is 8×8.

- **Fuzzy Morphological Top-Hat.** The fuzzy top-hat by closing, introduced in Sect. 2.4, is applied to the previous step as a curvilinear detector, in order to enhance the thin regions that are darker than their neighbouring pixels. We emphasize that, to achieve a low processing time, only one fuzzy top-hat operator is computed, in contrast with the curvilinear detector in Sect. 4.3.

  The fuzzy conjunction used is the nilpotent minimum t-norm, along with its R-implication, the Fodor implication function:

  $$T_{\mathrm{nM}}(x,y) = \begin{cases} 0, & \text{if } x + y \leqslant 1, \\ \min(x,y), & \text{otherwise,} \end{cases} \qquad I_{\mathrm{FD}}(x,y) = \begin{cases} 1, & \text{if } x \leqslant y, \\ \max(1-x,y), & \text{otherwise.} \end{cases}$$

  (5.1)

  Besides fulfilling several theoretical properties (see Table 2.1), these operators are fast to compute and have provided good results in similar problems [74]. The structuring element, along with the thresholds of the subsequent hysteresis (Fig. 5.5(F)), were selected experimentally, obtaining the best results with a flat structuring element of size 9×9.

- **Background Estimation.** The black padding background is detected to avoid any artifacts in this region. Given the RGB original image, we compute its pixel-wise luminance [75], Y, as:

  $$Y(x) = 0.2989 \cdot R(x) + 0.5870 \cdot G(x) + 0.1140 \cdot B(x), \text{ for all } x \in \mathbb{Z}^2.$$

  (5.2)

  Afterwards, we perform a sweeping from the vertical border of the image (assumed to be background) until we find a value greater than the mean luminance across the whole image, in which we assume that the retina starts. Given the good contrast of the luminance channel, this simple procedure provides very good results across all images.

  Finally, the object of interest is slightly shrinked. This operation is aimed at avoiding the border of the retina, where vessels are very infrequent but some artifacts may appear. To do

so, we use a fuzzy erosion with the same fuzzy implication function and structuring element that in the previous step.

- **Border Removal.** The pixels outside the retina mask are set to zero. To do so, we employ the result from the previous step to differentiate between background and retina. In the image generated by the fuzzy top-hat operator, all background pixels are set to zero.

- **Hysteresis.** Lastly, a hysteresis is applied to binarize the image while retaining connected components. The lower and higher thresholds have been experimentally set to 0.060 and 0.105 respectively. This last step provides the final vessel estimation.

The behaviour of these steps can be visually inspected in Figure 5.5. In it, the processing flow of a single image shows the effect and suitability of each step when processing retinal eye-fundus images.
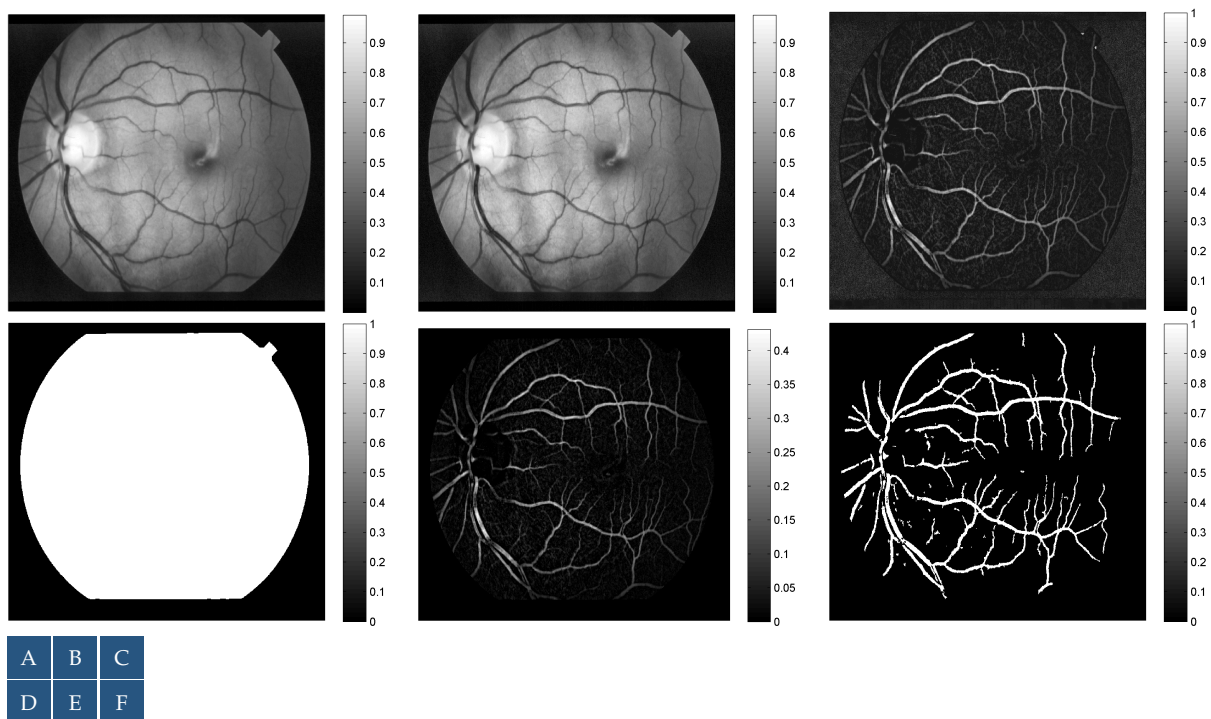
FIGURE 5.5: Intermediate steps of the real-time vessel segmentation method when processing the sample 235th from the STARE dataset: (A) green channel, (B) CLAHE, (C) fuzzy top-hat operator, (D) background mask, (E) border removal, (F) hysteresis.

We remark that the parameters used to obtain the segmentations are common to both DRIVE and STARE. They have been obtained by (i) selecting good-performance settings (for instance, a square-shaped structuring element instead of a circular-shaped one), and (ii) experimentally optimizing the numeric parameters (such as the hysteresis' thresholds). Such optimization has been done by maximizing the mean accuracy of the performance measures, since this provides a good trade-off between the amount of false positives and false negatives that our estimations contain. If we had optimized the parameters independently in each database, the performance measures would differ slightly with respect to the simultaneous optimization. In other words, the accuracy of the proposed method is not sensitive to small changes in the parameters. This behaviour, rarely found in the related literature [11, 9, 161], is a sign of the robustness of our method.

The algorithm is implemented in MATLAB® 2017b and C. All the code was implemented except for the CLAHE algorithm, for which we use the implementation by Zuiderveld [204].

5.3.2   Impact of the Parameters

Given the simplicity of the method, required in order to diminish its computational requirements, fine-tuning each of its steps acquires paramount importance. The different parameters that control the behaviour are the clip limit, the structuring element, the fuzzy conjunction and its residual fuzzy implication function used and the hysteresis thresholds.

The clip limit was set to 2 after empirical experimentation. Values within the range of 1.8 to 2.4 provide good results, both visually and in terms of the performance measures obtained with the whole method. We fixed it to 2 due to its robustness with regard to the whole set of images considered, both DRIVE and STARE samples.

On the other hand, the hysteresis lower and higher thresholds depend greatly on the preceding steps. A different structuring element provides very different gray level distributions. The hysteresis' thresholds are, then, obtained after fixing the remaining parameters: the structuring element, the conjunction and its corresponding residual implication function that control the behaviour of the fuzzy black top-hat transform. Once they are set, the thresholds are obtained as the ones that yield, in average, better performance measures for both datasets. We acknowledge that using the samples to tune parameters may bias its results. However, we remark that splitting the images into disjoint sets would yield a very small test set, increasing the bias of the performance measurements. In addition to it, the parameters are robust, so they should be very similar regardless of which images are used to compute them.

The most fundamental parameters appear to be the ones influencing the fuzzy top-hat operator. They control the geometrical and photometrical features of the objects that are enhanced, and, afterwards, extracted. The vast amount of different conjunctions, fuzzy implication functions and structuring elements make this step a very versatile one. For one thing, the conjunction and fuzzy implication function used were selected as the nilpotent minimum and its R-implication, the Fodor implication function. Other pairs of t-norms and R-implications (such as the ones in Table 2.1, page 11) also appear to work well in our situation. However, the nilpotent minimum was chosen due to the good results obtained with it in other image processing applications, such as edge detection [34].

The structuring element is defined by both its shape and size. The former affected the pixels being segmented and, thus, the performance measures. Due to the nature of the problem, only isotropic shapes were considered as structuring elements. More specifically, we considered several 2D Gaussian functions scaled so their maximum value is 1. As can be observed in Fig. 5.6, fewer artifacts were segmented with higher-variance Gaussian functions.

We tested Gaussian-shaped structuring elements with different variances. For each of them, the segmentation was obtained with the hysteresis threshold that provided the best performance measures. At a certain point, approximately for $\sigma = 75$ pixels, the segmentation reaches a steady response, as can be observed in Fig. 5.7(A). As a matter of fact, with structuring elements corresponding to variances larger than $\sigma = 75$ the segmentations provided are almost identical, as shown in Fig. 5.7(A). The shape of the structuring element was, finally, set to a flat surface (equivalently, a Gaussian function with infinite variance) given its robustness and ease of use.

Varying the size of the structuring element, on the other side, affected both the performance measures and the computational complexity of the method. Once the shape was fixed to a flat structuring element, several sizes were considered and compared, as observed in Fig. 5.7(B). The computational complexity always increases as the size of the structuring element increases, whereas the best performance measures are clearly bounded. Similarly to the case of varying the shape of the structuring element, each size was also used along with the best thresholds of the hysteresis, so their comparison becomes fair. The trade-off we preferred was selecting a structuring element of size 9×9.

With these settings, the best hysteresis thresholds were $t_{\text{low}} = 0.060$ and $t_{\text{high}} = 0.105$.
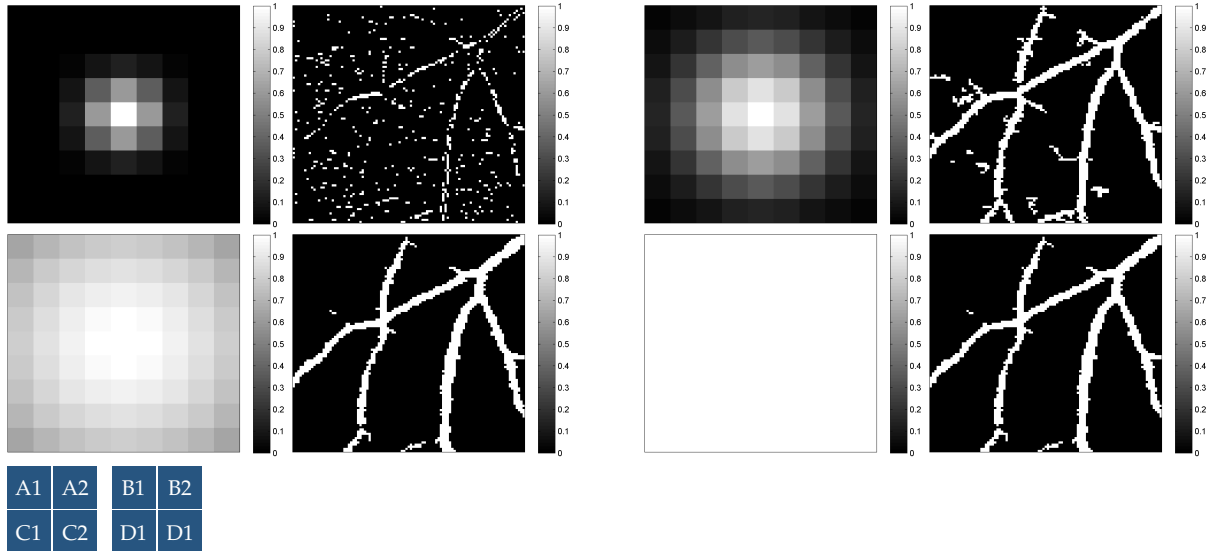
| A1 | A2 | B1 | B2 |
| C1 | C2 | D1 | D1 |

FIGURE 5.6: (1) Gaussian-shaped structuring elements with different variance σ^2, and (2) the segmentation they provide for the 4th sample of the DRIVE dataset, $(x, y) \in [100, 200].^2$. (A) $\sigma^2 = 1.0$. (B) $\sigma^2 = 4.2$. (C) $\sigma^2 = 75.0$. (D) $\sigma^2 = \infty$.

| A | B |
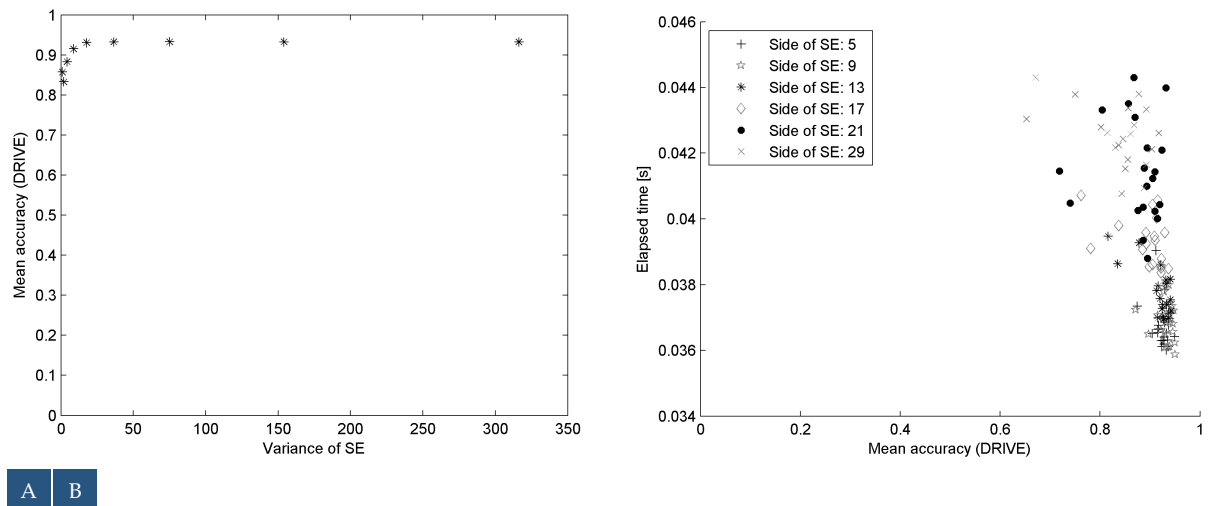
FIGURE 5.7: Performance of the method for different structuring elements on the DRIVE database. (A) shows the mean accuracy varying the variance of the Gaussian function that represents the structuring element. (B) represents executions on different samples, where the marker type indicates the size of the structuring element.

5.4   Results

In this section we measure the results provided by our real-time algorithm according to several pixel-wise performance measures, summarized in Table 3.3. Besides, they are compared with those of other state-of-the-art methods, some of which were already compared in Table 3.4.

Comparing algorithms and determining which approach provides the best segmentation is a hard task. The notion of *best method* could be defined as the one that provides more information towards solving a major goal such as automatic diabetic retinopathy detection. However, due to the complexity of such comparison, we use other indicators. Visual inspection, as the one shown in Fig. 5.8, may be useful, but falls short when comparing similar methods or when the task is rather subjective, which is our case. As in Sect. 3.4, to provide a plausible yet accurate comparison between methods we rely on public datasets, objetive metrics and statistical tests.

Firstly, we use the DRIVE [162] and STARE [83] databases. Secondly, each segmentation is assigned a series of numeric values assessing the similarity of such segmentation with the ground

A1	A2	B1	B2
C1	C2	D1	D2
E1	E2	F1	F2

FIGURE 5.8: (1) Visual comparison of fast methods with the 14th sample of DRIVE, and (2) a detail corresponding to $x, y, \in [100, 200]^2$ (upper left corner). (A) 1st Ground Truth: Acc: 1.00, Pre: 1.00. (B) 2nd Ground Truth: Acc: 0.95, Pre: 0.83. (C) our method (0.039 seconds)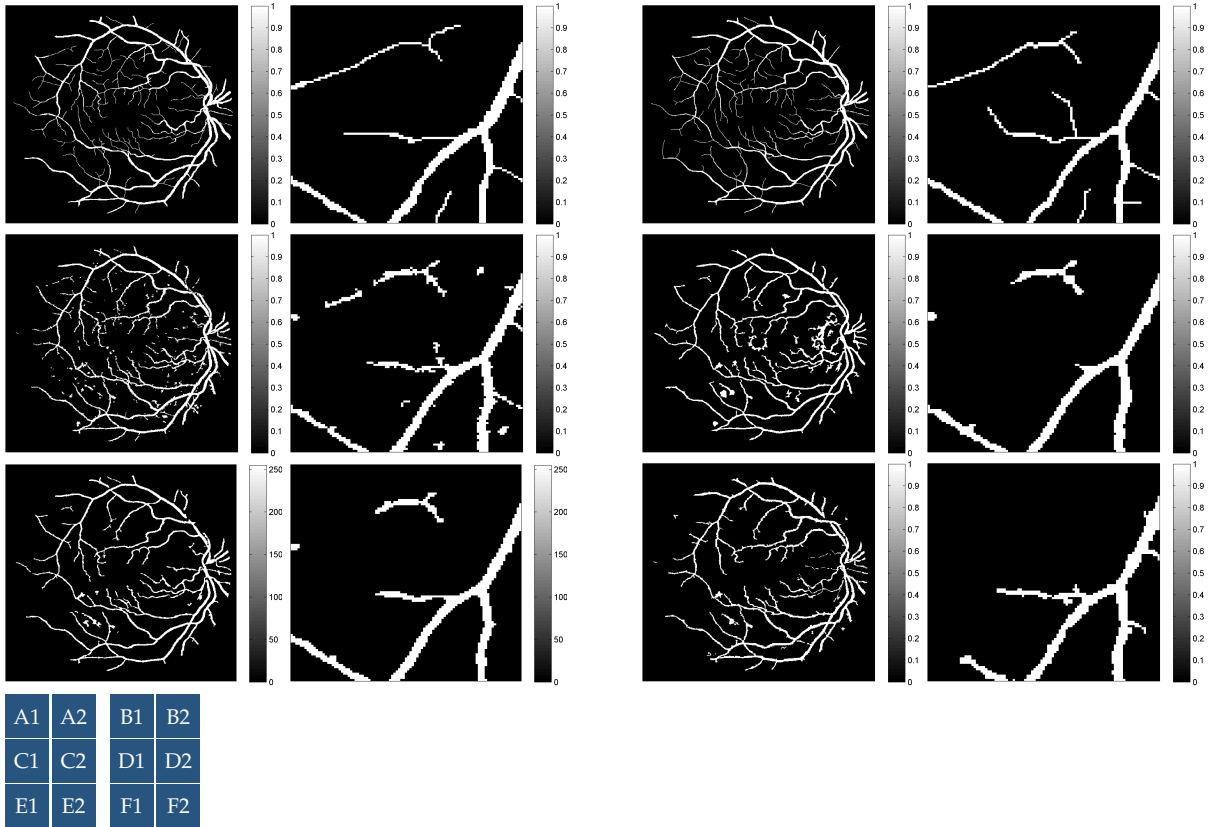: Acc: 0.94, Pre: 0.72. (D) Bankhead [15] (0.069 seconds): Acc: 0.93, Pre: 0.67. (E) Arguello [9] (0.420 seconds): Acc: 0.95, Pre: 0.77. (F) Roychowdhury [149] (2.45 seconds): Acc: 0.94, Pre: 0.72.

truth. Lastly, each method provides a series of results, one per image, that can be thought of as a statistical sample. They can be compared by means of a statistical test to answer the question of whether one method provides better results than another one with a certain statistical significance. In our case, we use the Unilateral Wilcoxon Signed-Rank test [189] with significance level $\alpha = 0.05$, as in Sect. 3.4.1. To recall, this test compares the hypothesis of whether the median of the difference of one distribution minus the other is greater than zero or not. Although this test compares the median instead of the mean, it is used due to its robustness against non normality, which is present in our distributions.

Both DRIVE and STARE are used to compare the methods. The performance measures of different methods over these two benchmarks are shown in Tables 5.2 and 5.3. We remark that these tables only depict the methods whose source code or segmentations were available so that metrics are computed uniformly. In addition to it, the background pixels (provided by the authors for the DRIVE dataset, computed for the STARE samples) have not been taken into account to compute the performance measures.

When executed in a single-core CPU, the fastest method is the one we introduced in Sect. 5.3. The performance measures that it presents are also remarkable, being in some situations statistically better than the one provided by the second expert, especially for the measures of specificity and precision. Besides, the metrics are balanced (*i.e.* there is no metric with very low values, in contrast to, for instance, the method by Soares with LMSE). The best sensitivity metrics, specially for the STARE database, are the ones provided by Azzopardi, whereas several methods, including ours, present a high accuracy.

Regarding the time per sample, an extensive comparison is provided. Whenever possible, the

TABLE 5.2: Results of methods with the 20 test samples of the DRIVE database [162], ordered by execution time, compared against our results, in bold typography. The superscripts should be interpreted as in Table 5.1. * indicates that the authors also executed their method in CUDA® compatible GPU cards, with which they required in average 14 ms per sample without taking into account the associated overhead, whose impact is specially important when processing images one by one.

Method	Time per sample	System	Accuracy	Sensitivity	Specificity	Precision
2nd ground truth	—		0.947† (0.005)	0.776† (0.059)	0.972$^\sim$ (0.008)	0.807$^\sim$ (0.044)
Our method	0.037 s (0.008 s)	3.10 GHz, 16 GB RAM	**0.938** (0.007)	**0.721** (0.069)	**0.970** (0.012)	**0.786** (0.065)
Bankhead [15]	0.066 s (0.004 s)	3.10 GHz, 16 GB RAM	0.931‡ (0.008)	0.745† (0.047)	0.958‡ (0.008)	0.723‡ (0.050)
Alonso-Montes [5]	0.193 s	SIMD parallel processor	0.918	-	-	-
Argüello [9]	0.421 s* (0.001 s)	3.10 GHz, 16 GB RAM	0.943† (0.006)	0.721$^\sim$ (0.061)	0.976† (0.008)	0.815† (0.049)
Chanwimaluang [48]	1.830 s (0.045 s)	3.10 GHz, 16 GB RAM	0.916‡ (0.015)	0.671‡ (0.082)	0.952‡ (0.026)	0.686‡ (0.084)
Roychowdhury [149]	2.45 s	2.60 GHz, 2 GB RAM	0.936‡ (0.005)	0.682‡ (0.047)	0.973$^\sim$ (0.007)	0.786$^\sim$ (0.048)
Roychowdhury [148]	3.11 s	2.60 GHz, 2 GB RAM	0.936‡ (0.004)	0.717$^\sim$ (0.057)	0.968$^\sim$ (0.008)	0.767$^\sim$ (0.052)
Azzopardi [11]	3.979 s (0.116 s)	3.10 GHz, 16 GB RAM	0.945† (0.005)	0.762† (0.080)	0.972$^\sim$ (0.012)	0.803$^\sim$ (0.063)
Soares (LMSE) [161]	8.733 s (0.296 s)	3.10 GHz, 16 GB RAM	0.929‡ (0.009)	0.497‡ (0.048)	0.992† (0.003)	0.898† (0.038)
Soares (GMM) [161]	9.592 s (0.157 s)	3.10 GHz, 16 GB RAM	0.947† (0.006)	0.728$^\sim$ (0.047)	0.979† (0.005)	0.838† (0.037)
Soares (kNN) [161]	28.542 s (0.182 s)	3.10 GHz, 16 GB RAM	0.940$^\sim$ (0.007)	0.728$^\sim$ (0.046)	0.972$^\sim$ (0.006)	0.789$^\sim$ (0.041)
Staal [162]	15 min	1.00 GHz, 1 GB RAM	0.944† (0.006)	0.719$^\sim$ (0.070)	0.977† (0.009)	0.825† (0.053)
Zana [199]	-	-	0.943† (0.008)	0.669‡ (0.077)	0.983† (0.008)	0.858† (0.051)

same system has been used to compute all segmentations. As in Sect. 5.2, the system used is an Intel® Core™ i5-3340 quad core @ 3.10 GHz with 16 GB of RAM, running Windows 10 Pro and MATLAB® R2017b. When such computation has not been possible, the time per sample is provided by the authors and using up-to-date hardware at their respective publication time [149, 15].

The only methods that could be used for real-time video processing are the one by Bankhead (12 − 15 frames per second) and our method (18 − 27 frames per second). We remark that these frame rates have been obtained with a single-core CPU execution. Given the ease of parallelization of different frames, with the current quad core system they could be sped up around four times. We also remark that Argüello optimized their method with CUDA® to be executed in compatible GPU cards. We have measured the execution time of their method based on the author's CPU implementation in order to make the comparison fair, since all the other methods are only implemented in CPU. The parallel nature of most of the methods, including ours, would provide similar speed up factors if a GPU implementation was developed.

Some methods improve the objective metrics of our algorithm, being the method by Azzopardi *et al.* [11] the one that provides better results when taking into account all metrics. However, none of the methods that significantly improve the segmentations provided by our approach is comparable in terms of its execution time, making them unsuitable for, among other uses, pre-screening

TABLE 5.3: Results of methods with the STARE database [83] ordered by execution time, compared against our results, in bold typography. The superscripts should be interpreted as in Table 5.1. * indicates that the authors also executed their method in CUDA® compatible GPU cards, with which they required in average 18 ms per sample.

Method	Time per sample	System	Accuracy	Sensitivity	Specificity	Precision
2nd ground truth	—		0.952~ (0.005)	0.895† (0.108)	0.956‡ (0.018)	0.642‡ (0.068)
Our method	0.057 s (0.012 s)	3.10 GHz, 16 GB RAM	**0.954** (0.011)	**0.647** (0.214)	**0.978** (0.005)	**0.737** (0.106)
Bankhead [15]	0.081 s (0.004 s)	3.10 GHz, 16 GB RAM	0.947‡ (0.010)	0.812† (0.044)	0.959‡ (0.011)	0.617‡ (0.093)
Argüello [9]	0.547 s* (0.002 s)	3.10 GHz, 16 GB RAM	0.959† (0.009)	0.730† (0.169)	0.978~ (0.011)	0.748† (0.093)
Chanwimaluang [48]	2.405 s (0.045 s)	3.10 GHz, 16 GB RAM	0.935‡ (0.023)	0.740† (0.094)	0.951‡ (0.026)	0.579‡ (0.116)
Roychowdhury [149]	3.95 s	2.60 GHz, 2 GB RAM	0.962† (0.010)	0.676~ (0.069)	0.985† (0.007)	0.790† (0.079)
Azzopardi [11]	5.105 s (0.121 s)	3.10 GHz, 16 GB RAM	0.954~ (0.014)	0.819† (0.082)	0.965‡ (0.015)	0.671‡ (0.095)
Roychowdhury [148]	6.7 s	2.60 GHz, 2 GB RAM	0.942‡ (0.005)	0.694~ (0.076)	0.962‡ (0.014)	0.610‡ (0.066)
Soares (LMSE) [161]	10.894 s (1.893 s)	3.10 GHz, 16 GB RAM	0.946‡ (0.013)	0.412‡ (0.074)	0.991† (0.005)	0.781† (0.114)
Soares (GMM) [161]	11.594 s (0.186 s)	3.10 GHz, 16 GB RAM	0.945~ (0.024)	0.651~ (0.070)	0.969~ (0.023)	0.670~ (0.145)
Soares (kNN) [161]	37.777 s (0.548 s)	3.10 GHz, 16 GB RAM	0.961† (0.005)	0.676~ (0.084)	0.985† (0.007)	0.782† (0.084)
Hoover [83]	5 min	SPARCstation 20	0.946‡ (0.007)	0.673~ (0.055)	0.969~ (0.011)	0.644‡ (0.109)

purposes or real-time response. When comparing real-time algorithms executed in CPU, the proposed one achieves the best results in accuracy, specificity and precision.

5.5 Discussion

We conclude this chapter with a brief account of the main points of this work. We include some remarks regarding the strengths and weaknesses of our proposed real-time vessel segmentation algorithm, which represents one of the main contributions of this thesis.

In this chapter, we have focused on the problem of retinal vessel segmentation. This, which can help ophtalmologists to detect eye diseases, is a tedious task when it is carried out manually. First, we have developed an abstract implementation based on a curvilinear object detector. With it, we have compared how appropriate is each of the detectors introduced in Ch. 4. Second, after realizing that the fuzzy top-hat operator fits very well into this task, we have designed a real-time retinal vessel segmentation based on it. To do so, we have tailored the original algorithm to lower its processing time and, at the same time, take the most of the fuzzy top-hat operator—which is the most time-consuming task of the whole algorithm. With the fuzzy black top-hat, we deal efficiently with the uncertainty and vagueness present in these images, while leveraging the geometric and photometric features of vessels presented in Sect. 5.1 as much as possible.

The optimized method, which is presented in Sect. 5.3, is composed of several fast steps. The CLAHE algorithm processes the green channel of RGB eye fundus images. This pre-processing step enhances the contrast, does not add noise and makes all images homogeneous. Without this procedure, the performance drops at least 2% with regard to the accuracy of the whole dataset,

but around 12% regarding its sensitivity. The fuzzy black top-hat is then applied to extract vessels. After removing the outer padding, the image is binarized with a hysteresis. The parameters that control each one of these steps have been obtained experimentally with a sistematic method, as shown in Sect. 5.3.

The results obtained by our method position it among the state of the art, as reflected in Tables 5.2 and 5.3. Since visual inspection is a subjective comparison, several performance metrics are computed: accuracy, sensitivity, specificity and precision. Our algorithm presents very high values in all performance metrics, specially when compared to other real-time methods, which are infrequent in the literature. Besides, we have executed all the methods at our reach to provide a fair comparison between their computational requirements in practice. We emphasize, as developed in the rest of the chapter, that our focus is on the speed of the method, which makes it easily embeddable into a high-complexity diagnosis solution.

The method that we have introduced is an example of how few, well-selected operations can provide good results in a task which, a priori, seems hard to solve. Thus, the method not only becomes valuable due to its results, but also due to its robustness: the same parameters work well across two datasets with different characteristics. In short, its simplicity brings along both fast processing times and ease of generalization.

This method is based on the curvilinear detector from Sect. 4.3, which has been highly optimized. In particular, several techniques have been used to reduce the computation time. Firstly, it is programmed in C code. Secondly, we purposely chose structuring elements with very few unique values—only two unique values in the case of flat structuring elements. This makes feasible the precomputation of conjunctions and implications, which greatly increases the efficiency of morphological operations. Lastly, all steps have been designed to be extremely fast: the hysteresis is performed by traversing the image twice, whereas the rest of steps are local and can be parallelized.

Although the processing time is excellent for medium-sized images, processing high-resolution video would require faster methods. For instance, the time that our implementation requires to process a $2,336 \times 3,504$ pixels image is between 1.53 and 1.61 seconds. Some strategies to increase the computational time are multi-core execution, GPU optimization [9] or downsampling the images [15]. These would make our method very suitable to pre-screen large datasets in a short amount of time or to generate a good vessel estimator nearly instantaneously. Providing this first pre-screening, a more demanding algorithm could be executed to provide a final answer, which would only need to process a fraction of the original samples.

Chapter 6

Soft Color Morphology

In this chapter we extend the fuzzy mathematical morphology operators, introduced in Sect. 2.4, to process multivariate images. We provide a general definition, for images of any color space and spatial dimension. They have been designed taking into account a very important case: natural images. Whenever they are encoded using the CIELab color space, our operators provide interpretable results. The soft color morphological operators are fast and provide smooth outputs from a visual point of view. They can be easily customized while also being highly interpretable, which is why they can be very helpful tools when dealing with some computer vision challenges. As a matter of fact, they are used in Chapter 7 to deal with a real application, hair removal in dermoscopy images, obtaining very competitive results.

The problem of extending the morphological operators to color images is studied in Sect. 6.1, along with the state of the art. In Sect. 6.2 we introduce the basic operators of the soft color morphology, which are studied from a theoretical point of view in Sect. 6.3. They are extensively compared with other morphological paradigms for color images in Sect. 6.4. We finish the chapter with an analysis of the operators and their behaviour in Sect. 6.5.

6.1 Mathematical Morphology and Color

Binary mathematical morphology was promptly extended to grayscale images [80]. One of such examples is the fuzzy mathematical morphology, introduced in Sect. 2.4. However, dealing with natural color images or with multivariate images has proved to be more challenging. In this section, we study why this generalization is so hard and the approaches already proposed in the literature.

6.1.1 The Problem of Color Ordering

Mathematical morphology requires the color space to admit an order or, at least, a supremum and infimum operation. The lattice-based definition of erosion (or dilation) is based on using the structuring element to select some pixels in a neighbourhood, and then selecting the minimum (or maximum) of the colors that such pixels contain. From a higher level perspective, we interpret the dilation as the operation of enlarging a region that represents an object. To do so, the dilation *prioritizes* the colors of such object over the colors of its surrounding background. Taking into account either point of view, and regardless of the color space used, the colors must be comparable somehow.

Although binary and grayscale values can be ordered easily, multivariate color spaces may not admit a *consistent* order: different prior knowledge of the task at hand may suggest different color orderings [181]. For instance, colors from the RGB color space have three components. More concretely, a red pixel and a blue pixel may not be comparable, neither have a supremum or infimum that is always consistent with human perception.

Besides not necessarily having a consistent order, we encounter what is known as the false color problem: when processing color images, some pixels may get assigned colors that were not present

in the original image. This is sometimes described as a problem to overcome, although it may not necessarily be an undesired situation [77]. For instance, the fuzzy mathematical morphology (see Sect. 2.4) comprises operators that may output pixels whose value did not appear in the original image. However, the false color problem is more critical for multivariate images, due to, among other factors, the correlation between channels. In other words, providing only pixels whose value was present in the original image may be a desirable property for such multivariate image. For instance, a corregistered pair of PET and CT scans may not have the same inter-channel correlations as a natural RGB image.

The vast majority of color morphology paradigms can be divided into component-wise and vector approaches [6]. The former ones split the image in channels, that are processed independently. On the other hand, vector approches process all channels simultaneously. To do so, they must order the vectors that represent colors, for which several alternatives have been proposed [6, 17]:

- Marginal ordering: sorting according to one component. For instance, when considering different orders, one per component.

- Conditional ordering or lexicographical ordering: single-channel ordering applied sequentially to all components in a specific order, in which ties are resolved by the next component.

- Partial ordering: sorting the data according to partitions in the sample space. Each partition is assigned a number that is used to compare its elements with samples from other partitions. It takes into account the structure of the vectors in the space, such as geometric relations in the sample space [6].

- Reduced ordering: sorting with an overall combination of all components. It can be thought as a function that maps color vectors to real values, which are used to compare the original multivariate data. If the function is not injective, the associated relation between vectors is not a total order, but a pre-order [6]. This is the most general approach to define a total order for multivariate data.

Besides, orderings may also be adaptive if they change depending on the original data. In Sect. 6.1.2 we list several works that use these orders, emphasizing their strengths and shortcomings.

The color ordering that considers our soft color morphology operators also considers the spatial location of such colors. In short, it favours colors with a higher first component (when considering CIELab, bright colors), and with less priority the color of close pixels. It is formally specified in Def. 6.1 (see Sect. 6.2).

6.1.2   State of the Art

Several works have been proposed to extend mathematical morphology to color or multivariate images. However, they are scarcely used by practitioners. In other words, there is no successful definition for color mathematical morphology operators that is commonly accepted.

The component-wise approach, such as the one presented by Gu *et al.* [78], can be seen as a marginal ordering in each component taken independently. It considers a multivariate image as a stack of grayscale layers and processes each of them independently. It does not exploit spectral correlations and it is prone to introducing new colors.

The vector approach to color morphology, on the other hand, encompasses the methods that process all the channels of the image simultaneously. Sometimes, they specify a total order to then apply the dilation and erosion according to the lattice-based definitions of erosion and dilation [158], avoiding thus the creation of new colors. Aptoula and Lefèvre [7] explore the lexicographical ordering with the RGB and HSL color spaces. This approach resolves almost all orders taking into

account one channel, which we must expect to separate well the objects to be transformed. Chanussot and Lambert [47] present a variation of the lexicographical ordering, in which they employ an 8-bit RGB representation of images. In this case, the higher bits of each channel are given more importance than the lower bits of the same or other channels. Although the behaviour provided by this ordering is more aligned with human perception, it is arbitrary and not necessarily meaningful. For instance, the human eye can distinguish more green shades than red ones. Moreover, the channels of hyperspectral imagery may have different statistical distributions which makes them incomparable. Another variant of the lexicographical order is employed by Louverdis *et al.* [116], where the HSV color space is used, ordering the colors by minimizing their value, then maximizing their saturation and finally minimizing their hue. Sartor and Weeks [153] present a method that orders colors based on their distance with respect to a reference color, being thus a reduced ordering. This approach provides results that match the expectations, specially when the reference color has been carefully selected. Its drawbacks are the need of such user-supplied color (which could also be thought as a desirable customization) and the theoretical instability: two very distant colors may change their precedence due to small perturbations. Witte *et al.* [56] employ the fuzzy mathematical morphology operators with multivariate images by extending the usual operations. Specifically, they extend the notions of minimum, maximum, addition, negation and product for colors encoded in several color spaces—RGB, HSV and CIELab. That is, instead of considering a color ordering, they consider operations between colors. Accordingly, they consider structuring elements encoded in the same color spaces. Bouchet *et al.* [41] use a fuzzy order to create a total order between RGB triplets. It computes an aggregated value for each color, being thus a reduced ordering approach. For each channel, they learn a pixel-wise fuzzy preference relation. The three fuzzy preference relations are then aggregated with the arithmetic mean. Valle and Valente [178] propose a total order in the CIELab color space, based on the distance with respect to a reference color and their relative position. It can be seen as a conditional ordering in a CIELab-related color space. They propose a binary operation between colors in the CIELab complete lattice that is associative, commutative and has a neutral element. Employing it, they define morphological operators that admit structuring elements also encoded in the CIELab color space.

A series of works also explore adaptive orderings. All of them follow the same strategy than the previous algorithms—finding a total order between colors—but they do so in an image-dependent fashion. Velasco and Angulo [181] study and compare several supervised and unsupervised methods to reduce the dimensionality of the color space: principal component analysis, support vector machine, reference color and others. These approaches typically assume that small objects are located on the foreground, whereas large objects conform the background. One of such approaches, also by Velasco and Angulo [182], is the random projection technique: they employ a one-dimensional measure of the degree of centrality of a sample—known as a depth function—, and maximize it for each possible projection of the multivariate data into a one-dimensional space. That is, they assign each color to a single real value, being thus a reduced ordering. Benavent *et al.* [28] also consider an adaptive partial ordering, which in this case is based on the image histogram. By considering a smoothed version of the histogram, they obtain a measure of how frequent a certain color is within the image. To decrease the computational complexity, they average the histograms of multiple images to obtain a reference color ordering for new, similar images. Lézoray [109] introduced a method that learns a color ordering from the image, being thus an adaptive reduced ordering. It does so by learning a rank transformation on a complete lattice—which is equivalent to a total order. Laplacian eigenmaps are used to learn a non-linear projection of the color space.

6.2 Soft Color Morphology Operators

We now formally introduce the *Soft Color Morphology* basic operators. To do so, we extend the fuzzy mathematical morphology from Sect. 2.4 to deal with multivariate images. In short, our

operators work by ordering using mainly the first channel, maintaining (and spreading) the values of the rest of the channels accordingly.

We define here a method to totally order the colors of an image in a specific neighbourhood, which is used to define a **max** and **min** of a set of pixels. Let us assume we are processing an image A at the location y, and that we are comparing some colors $A(x)$ for x in the neighbourhood of y. We are interested in the information contained in the first channel, and so we first order the spatial locations x by the *first component* of their color, $A_1(x)$. To deal with ties, which are uncommon in natural images, we first select the spatial location x which is *closest*—with regard to the Euclidean distance—to the location of the pixel we are processing, y. Further ties are resolved with the lexicographical order. That is, ordering by the first channel, then by the second, and so on. The following definition formally introduces these **max** and **min** operators.

Definition 6.1. — *Let $A : \mathbb{Z}^n \to \mathfrak{C}$ be a multivariate image. Let us consider a location $y \in \mathbb{Z}^n$ and a neighbourhood of y, $N \subset \mathbb{Z}^n$. Then, the maximum and minimum of the colors $A(x)$ with respect to y and N, $\max\limits_{x \in N}^y (A)$ and $\min\limits_{x \in N}^y (A)$, are defined as*

$$
\begin{cases}
\nabla_1 & = & \arg\max\limits_{x \in N} A_1(x), \\
\nabla_2 & = & \arg\min\limits_{x \in \nabla_1} d_2(x,y), \\
\max\limits_{x \in N}^y (A) & = & \max\limits_{x \in \nabla_2}^{lexic} A(x),
\end{cases}
\qquad
\begin{cases}
\Delta_1 & = & \arg\min\limits_{x \in N} A_1(x), \\
\Delta_2 & = & \arg\min\limits_{x \in \Delta_1} d_2(x,y), \\
\min\limits_{x \in N}^y (A) & = & \min\limits_{x \in \Delta_2}^{lexic} A(x),
\end{cases}
$$

where $d_2 : \mathbb{R}^2 \to \mathbb{R}$ is the Euclidean distance and $\max^{lexic}$ and $\min^{lexic}$ are the lexicographic maximum and minimum.

We emphasize that this ordering, in which ties are first resolved according to the spatial location, is not appropriate to define idempotent opening and closing operations. However, any other ordering would introduce some bias towards a particular color—and we consider that there is not a consistent order among colors. Besides, resolving ties according to the spatial location is needed to obtain the full color preservation property (see Proposition 6.9).

Once we have defined an order between colors, we can proceed to define the soft color dilation and soft color erosion. They are the soft color counterparts to the fuzzy dilation and fuzzy erosion from Defs. 2.18 and 2.19. They are not defined using the lattice-based definition of morphological operators, but extending the fuzzy mathematical morphology definition to multivariate images.

Definition 6.2. — *Let I be a fuzzy implication function, $A : \mathbb{Z}^n \to \mathfrak{C}$ a multivariate image and $B : \mathbb{Z}^n \to [0,1]$ a structuring element. Then, the soft color erosion of A by B, $\mathcal{E}_I(A,B)$, is*

$$
\mathcal{E}_I(A,B)(y) = \min\limits_{x \in d_A \cap T_y(d_B)}^y \left\{ \Big(I\big(B(x-y), A_1(x)\big), A_2(x), \dots, A_m(x) \Big) \right\}.
$$

Similarly, we define the soft color dilation as follows.

Definition 6.3. — *Let C be a conjunction, $A : \mathbb{Z}^n \to \mathfrak{C}$ a multivariate image and $B : \mathbb{Z}^n \to [0,1]$ a structuring element. Then, the soft color dilation of A by B, $\mathcal{D}_C(A,B)$, is*

$$
\mathcal{D}_C(A,B)(y) = \max\limits_{x \in d_A \cap T_y(d_B)}^y \left\{ \Big(C\big(B(x-y), A_1(x)\big), A_2(x), \dots, A_m(x) \Big) \right\}.
$$

Essentially, in the absence of ties, the definition selects the spatial location x at which $C(B(x - y), A_1(x))$ is maximum (of course, restricted to the appropriate domain, $x \in d_A \cap T_y(d_B)$). Once the location is selected, the other channels are simply dragged.

The soft color morphology operators interpret the first channel as a measure to indicate with which degree the pixels belong to the *foreground* or to the *background*. When processing natural images with the CIELab color space, the first channel is already very informative, containing the luminance of a color pixel and disregarding the chromatic information. Using such channel to discriminate whether the objects belong to the foreground or the background provides a generalization of the same procedure for grayscale and binary images, which is coherent with human visual perception. In specific applications, practitioners must necessarily provide this information in accordance with the task they are dealing with, since images from different problems can be of a really different nature. Besides the luminance channel L* in natural images, other examples are the green channel, in RGB-encoded retinal angiographies; or an artificial channel that orders the colours according to their frequency, such as in Benavent *et al.* [28]. Our operators are designed to easily handle these and similar cases.

Fig. 6.1 shows the soft color dilation and soft color erosion with the balloons image. As shown in this first example, the soft color dilation makes bright objects larger, while the soft color erosion diminishes them. We emphasize the behaviour when two objects clash: either one object is placed *over* the other or they collide into a well-defined border.

FIGURE 6.1: Soft color erosion (A), the L*a*b*-encoded *Balloons* image (B) and soft color dilation (C), with the minimum t-norm, its residual implication (the Gödel implication) and a 31×31-pixel Gaussian-shaped structuring element, with $\sigma = 8$ pixels and a maximum value of 1. We remark that the irregular shapes of the eroded balloons reflect their irregular illumination.

Given these two definitions, other morphological operators are defined similarly to the corresponding ones in binary or fuzzy mathematical morphology.

Definition 6.4. — *Let $A : \mathbb{Z}^n \to \mathfrak{C}$ be a multivariate image and let $B : \mathbb{Z}^n \to [0,1]$ be a structuring element. Let C be a conjunction and let I be a fuzzy implication function. Then, the soft color opening of A by B, $\mathcal{O}_{C,I}(A,B)$, is defined as:*

$$\mathcal{O}_{C,I}(A,B) = \mathcal{D}_C(\mathcal{E}_I(A,B), \overline{B}).$$

Definition 6.5. — *Let $A : \mathbb{Z}^n \to \mathfrak{C}$ be a multivariate image and let $B : \mathbb{Z}^n \to [0,1]$ be a structuring element. Let C be a conjunction and let I be a fuzzy implication function. Then, the soft color closing of A by B, $\mathcal{C}_{C,I}(A,B)$, is defined as:*

$$\mathcal{C}_{C,I}(A,B) = \mathcal{E}_I(\mathcal{D}_C(A,B), \overline{B}).$$

In Fig. 6.2, the soft color opening and soft color closing operators are visualized processing a 600×600-pixel dermoscopic image in the CIELab color space. In it, the skin lesion has a globular network pattern. The opening provides a darker image, in which small bright blobs disappear, but the big ones remain. On the other hand, the closing operator has a similar behaviour on the dark blobs, providing thus a brighter image.

In Fig. 6.3, we demonstrate the effect of using different t-norms. In it, the soft color dilation and soft color erosion operators are visualized processing a 600×400-pixel image in the CIELab
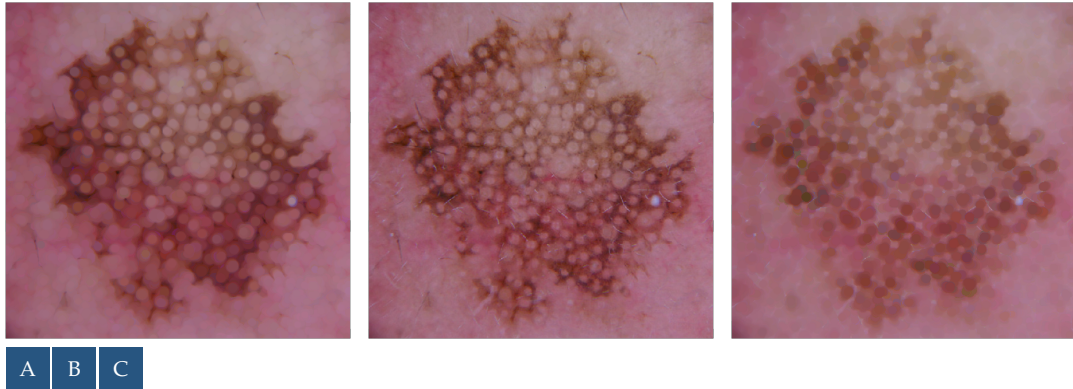
FIGURE 6.2: Soft color opening (A) and soft color closing (C) of a dermoscopy image (B) depicting a globular network, using the same configuration as in Fig. 6.1.

color space. Different t-norms are used to visualize their behaviour: the closer a t-norm is to the minimum operator, the greater the impact in the final result. The minimum t-norm and the Gödel implication provide a stronger response, whereas the Łukasiewicz pair provides the softer one. This is specially noticeable in high-contrasted regions, such as the grapes, and in regions that present texture, such as the pineapple. Although all examples are computed with a t-norm and its respective R-implication (see Table 2.1, page 11), this does not need to be always the case. More visual examples are found in Sect. 6.4.
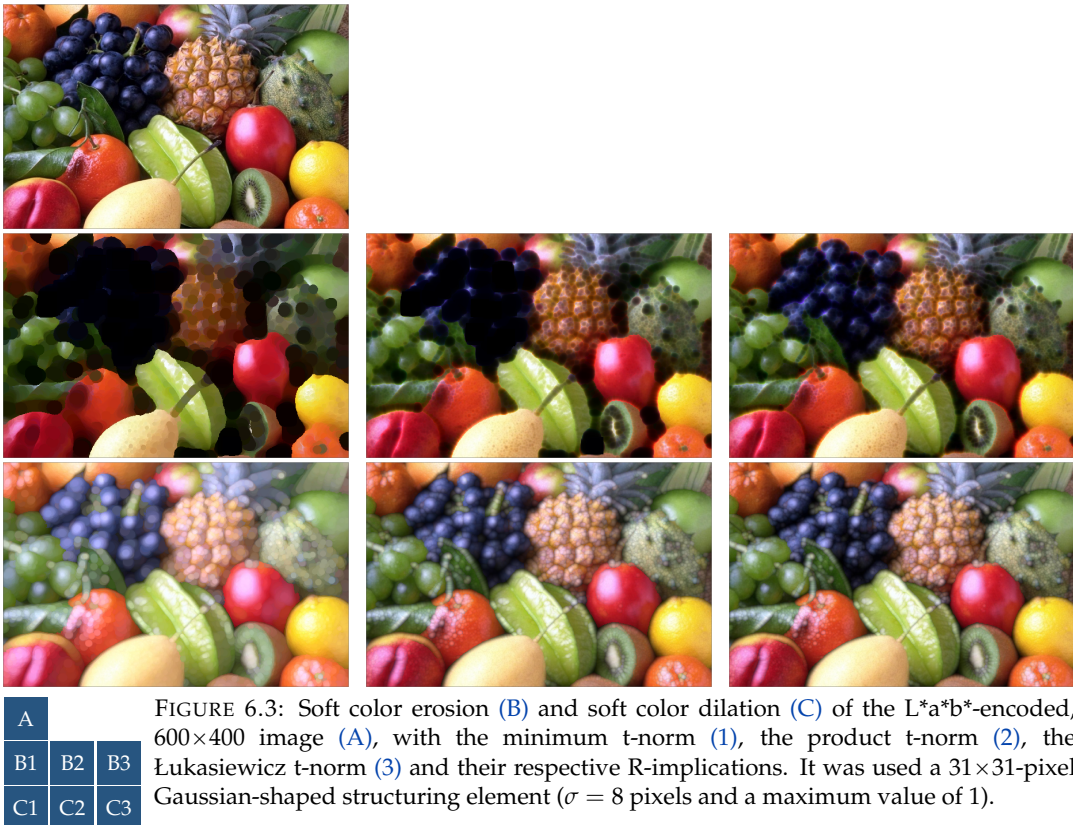

FIGURE 6.3: Soft color erosion (B) and soft color dilation (C) of the L*a*b*-encoded, 600×400 image (A), with the minimum t-norm (1), the product t-norm (2), the Łukasiewicz t-norm (3) and their respective R-implications. It was used a 31×31-pixel Gaussian-shaped structuring element ($\sigma = 8$ pixels and a maximum value of 1).

In Fig. 6.4, different structuring elements are used along with the minimum t-norm and the Gödel implication. In this 100×100-pixel patch of the `jellybeans` image, we can visualize how different structuring elements affect the erosion and dilation operations. Whereas the 1×1 impulsive structuring element leaves the image unaffected (both in the case of the erosion and dilation operators), crisp structuring elements highlight limits between objects and their shades. On the

other hand, non-binary structuring elements show a smoother response in such limits. Also, a circular structuring element can be better approximated with a Gaussian decayment than with a crisp boundaries.
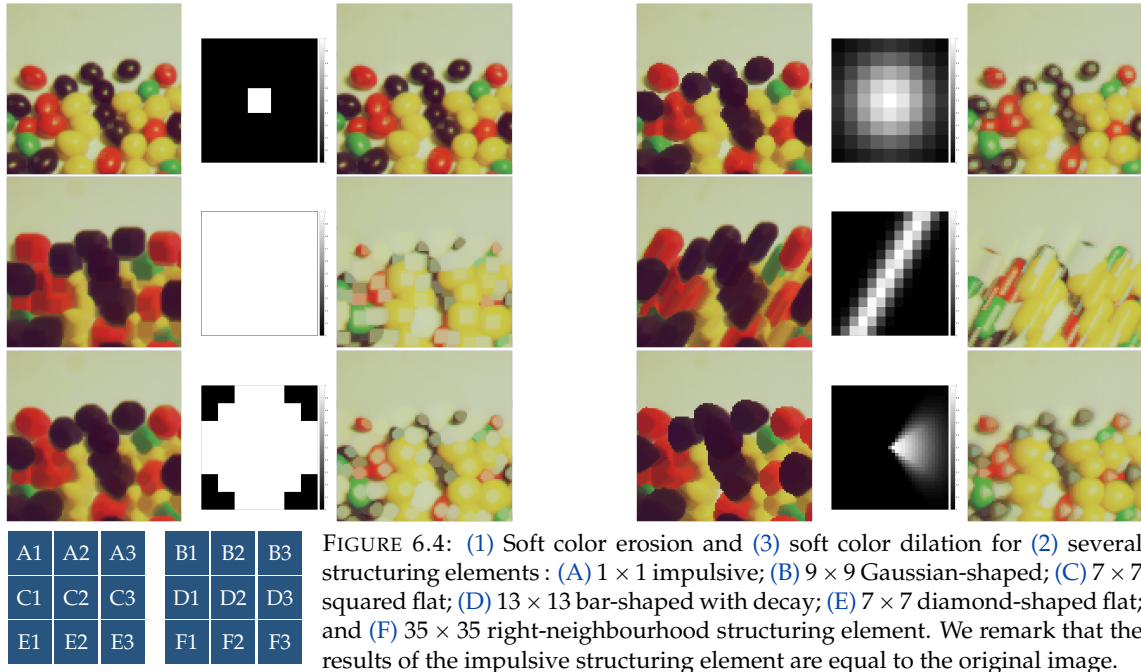

A1	A2	A3	B1	B2	B3
C1	C2	C3	D1	D2	D3
E1	E2	E3	F1	F2	F3

FIGURE 6.4: (1) Soft color erosion and (3) soft color dilation for (2) several structuring elements : (A) 1×1 impulsive; (B) 9×9 Gaussian-shaped; (C) 7×7 squared flat; (D) 13×13 bar-shaped with decay; (E) 7×7 diamond-shaped flat; and (F) 35×35 right-neighbourhood structuring element. We remark that the results of the impulsive structuring element are equal to the original image.

Besides, we can generalize the grayscale concept of pixel-wise difference between two images. Since we will be using the CIELab color space—which is perceptually uniform, see Sect. 2.2—, we can generalize such concept to the color framework by computing the pixel-wise Euclidean distance between images.

We use this fact to also generalize the top-hat operators and the morphological gradient operator in Defs. 6.6 and 6.7.

Definition 6.6. — *Let* $A : \mathbb{Z}^n \to \mathfrak{C}$ *be a multivariate image and let* $B : \mathbb{Z}^n \to [0, 1]$ *be a structuring element. Let C be a conjunction and let I be a fuzzy implication function. Let* $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *be the n-dimensional Euclidean distance.*

Then, the soft color top-hat by closing of A by B, $\mathcal{BTH}_{C,I}(A, B)$, *and the soft color top-hat by opening of A by B,* $\mathcal{WTH}_{C,I}(A, B)$, *are defined as:*

$$\mathcal{BTH}_{C,I}(A, B) = d\big(\mathcal{C}_{C,I}(A, B), A\big), \tag{6.1}$$

$$\mathcal{WTH}_{C,I}(A, B) = d\big(A, \mathcal{O}_{C,I}(A, B)\big). \tag{6.2}$$

Definition 6.7. — *Let* $A : \mathbb{Z}^n \to \mathfrak{C}$ *be a multivariate image and let* $B : \mathbb{Z}^n \to [0, 1]$ *be a structuring element. Let C be a conjunction and let I be a fuzzy implication function. Let* $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ *be the n-dimensional Euclidean distance.*

Then, the soft color morphological gradient of A by B, $\mathcal{G}_{C,I}(A, B)$, *is defined as:*

$$\mathcal{G}_{C,I}(A, B) = d\big(\mathcal{D}_C(A, B), \mathcal{E}_I(A, B)\big). \tag{6.3}$$

Also, the internal gradient of A by B, $\mathcal{G}_{C,I}^i(A, B)$, and the external gradient of A by B, $\mathcal{G}_{C,I}^e(A, B)$, are defined as:

$$\mathcal{G}_{C,I}^i(A, B) = d\big(A, \mathcal{E}_I(A, B)\big), \tag{6.4}$$

$$\mathcal{G}_{C,I}^e(A, B) = d\big(\mathcal{D}_C(A, B), A\big). \tag{6.5}$$

In Section 6.4, most of these operators are shown and are also compared with the corresponding operators derived from other morphological frameworks.

6.3 Properties

In this section we study the properties of the soft color morphology operators from a mathematical point of view.

6.3.1 Chromatic Preservation in CIELab

When dealing with natural images, it is natural to think that the erosion and dilation should not introduce colors whose shade or chromatic value was not originally present. The soft color morphological operators from Defs. 6.3 and 6.2 have been designed with this property in mind. Therefore, they preserve the chromatic information of colors when the CIELab color space is used.

Proposition 6.8. (Chromatic Preservation in CIELab) — *Let A be a CIELab-encoded image with spatial dimension n, let C be a conjunction and I a fuzzy implication function, and let B be a structuring element.*

Then, for all $y \in d_A$, there exist $x_1, x_2 \in d_A$ such that the components a^* and b^* of $\mathcal{D}_C(A, B)(y)$ and $\mathcal{E}_I(A, B)(y)$ are equal to, respectively, the components a^* and b^* of $A(x_1)$ and $A(x_2)$.

Proof. Recalling the definition of the soft color dilation, for a CIELab-encoded image,

$$\mathcal{D}_C(A, B)(y) = \max_{x \in d_A \cap T_y(d_B)}^{y} \left\{ \Big(C\big(B(x - y), L^*(x)\big), \, a^*(x), b^*(x) \Big) \right\}.$$

Thus, it is clear that the only channel that may have been modified is the first one: the combination $(a^*(x), b^*(x))$ previously existed (as a matter of fact, in the spatial location x). Since the chromatic information of a CIELab-encoded color is determined by the combination of the second and third components, a* and b*, and these channels are preserved, we conclude that such chromatic information is indeed kept.

The same argument holds for the soft color erosion. □

Of course, this property also holds for the closing and the opening operators, since they are combinations of dilations and erosions.

6.3.2 Full Color Preservation

In some settings, we should avoid creating colors that were not present in the original image—not only preserving the chromatic components, but all of them. As already introduced in Sect. 6.1, this is the so-called false color problem [77].

Our operators from Defs. 6.3 and 6.2 do preserve colors under certain conditions. Specifically, whenever we are using a binary structuring element and either a semi-norm (for the soft color dilation) or a fuzzy implication function that fulfills (NP) (for the soft color erosion). The following proposition provides sufficient conditions to enforce color preservation.

Proposition 6.9. (Full Color Preservation) — *Let $A : \mathbb{Z}^n \to \mathfrak{C}$ be a multivariate image and let $B : \mathbb{Z}^n \to [0,1]$ be a structuring element. Let C be a semi-norm and let I be a fuzzy implication function that fulfils (NP).*

If $B(x) \in \{0,1\}$ for all $x \in d_B$ and $B(0) = 1$, then $\mathcal{D}_C(A, B)$ and $\mathcal{E}_I(A, B)$ only contain colors that appear in A.

Proof. Let us see the case of the soft color dilation first. Since C is a semi-norm, $C(a, 1) = a$ and $C(0, a) = 0$ due to $C(0, a) \leqslant C(0, 1) = 0$ for all $a \in [0, 1]$. Thus, $C\big(B(x - y), A_1(x)\big)$ is either $A_1(x)$ or 0. If any of such values is different to zero, then the dilation in the location $y \in \mathbb{Z}^n$ is the maximum among those colors, all of which appear in A. Otherwise, if all values are 0, since $B(0) = 1$ we can derive that $A_1(y) = 0$ and, thus, $\mathcal{D}_C(A, B)(y) = A(y)$ due to the choice procedure in case of ties.

The soft color erosion has a similar proof. All fuzzy implication functions are increasing in the second variable, so $I(0, a) = 1$ for all $a \in [0, 1]$ since $I(0, a) \geqslant I(0, 0) = 1$ for all $a \in [0, 1]$. Since the fuzzy implication function fulfils the left neutrality principle (NP), $I(1, a) = a$ for all $a \in [0, 1]$. Thus, $I\big(B(x - y), A_1(x)\big)$ is either $A_1(x)$ or 1. The same argument that in the case of the soft color dilation concludes that the color provided was already present in the original image A. □

Again, the opening and closing do not create new colors provided the hypotheses of the proposition are satisfied. This is true given that they are just combinations of dilations and erosions, and neither of them can create new colors.

6.3.3 Generalization of the Fuzzy Mathematical Morphology Operators

Another interesting property of these operators is that they constitute a generalization of the fuzzy mathematical morphology ones [38] when applied to CIELab-encoded images. In other words, the soft color morphology operators, when applied to grayscale images encoded in CIELab, coincide with the corresponding operators from the fuzzy mathematical morphology paradigm using the same structuring element.

Let us recall the conversion between CIELab and grayscale colors from Def. 2.3. It is given by the maps π, ι, defined as:

$$\pi : \begin{array}{ccc} \text{L*a*b*} & \longrightarrow & [0, 1] \\ (l, a, b) & \longmapsto & l, \end{array} \qquad \iota : \begin{array}{ccc} [0, 1] & \longrightarrow & \text{L*a*b*} \\ G & \longmapsto & (G, 0, 0). \end{array}$$

Now, we can state the following proposition:

Proposition 6.10. (Generalization of Fuzzy Mathematical Morphology for grayscale images.) — *Let $A : \mathbb{Z}^n \to [0, 1]$ be a grayscale image and let $B : \mathbb{Z}^n \to [0, 1]$ be a structuring element. Let C be a conjunction and let I be a fuzzy implication function. Then,*

$$\mathcal{E}_I^{Fuzzy}(A, B) = \pi\big(\mathcal{E}_I(\iota(A), B)\big), \qquad \mathcal{D}_C^{Fuzzy}(A, B) = \pi\big(\mathcal{D}_C(\iota(A), B)\big).$$

where $\mathcal{E}^{Fuzzy}$ and $\mathcal{D}^{Fuzzy}$ correspond to the operators from Defs. 2.18 and 2.19.

Proof. To see this, we must prove that, when restricted to grayscale images, our new operators provide exactly the same results.

We begin by considering the soft color dilation. The CIELab conversion of A is $\iota(A)$, where $\iota(A)(x) = (A(x), 0, 0)$, and its support remains unchanged, $d_{\iota(A)} = d_A$.

The soft color dilation of $\iota(A)$ by B is then:

$$\mathcal{D}_C(\iota(A), B)(y) = \max_{x \in d_\iota(A) \cap T_y(d_B)}^y \left\{ \left(C\left(B(x-y), A(x)\right), 0, 0\right) \right\}.$$

Regardless the result of the choice procedure in case of ties, the grayscale projection of such image is:

$$\pi\left(\mathcal{D}_C(\iota(A), B)(y)\right) = \max_{x \in d_A \cap T_y(d_B)} \left\{ \left(C\left(B(x-y), A(x)\right)\right\},$$

which matches the definition of dilation from the fuzzy mathematical morphology paradigm. The case for the soft color erosion is proved in a similar way, and it holds for any structuring element B and fuzzy implication function I. $\qquad\square$

6.3.4 First Channel Monotonicity

The aforementioned generalization of the fuzzy mathematical morphology is an interesting property because it can be leveraged to prove several properties, such as the monotonicity of the soft color morphological operators. To argue about monotonicity, we must be able to compare both $\mathfrak{C}$-encoded colors and $\mathfrak{C}$-encoded images.

This motivates the following definition of h-order [76]:

Definition 6.11. — *Given a complete lattice $\mathcal{L}$ and a surjective mapping $h : \mathfrak{C} \to \mathcal{L}$, the h-order is the relation $\leqslant_h$ defined as:*

$$c_1 \leqslant_h c_2 \iff h(c_1) \leqslant h(c_2), \quad \forall c_1, c_2 \in \mathfrak{C}.$$

In the following, we order the colors according to the first component, induced by the map $\ell : \mathscr{C}_1 \times \ldots \times \mathscr{C}_m \longrightarrow \mathscr{C}_1$. In particular, when considering the CIELab as the color space to be used, the colors are ordered by their luminance. That is, for any CIELab-encoded color, (l, a, b), the map ℓ returns the luminance value: $\ell(l, a, b) = l$. The following definitions, although valid for any color space, base their nomenclature on the fact that the first channel contains the luminance—which, in particular, makes the naming accountable when using the CIELab color space.

Let us now consider the h-ordering given by the map ℓ, denoted as $\leqslant_\ell$. For any two colors $c_1, c_2 \in \mathfrak{C}$ we say that c_1 is *lighter or equal than* c_2 when $c_1 \leqslant_\ell c_2$. That is, if and only if $\ell(c_1) \leqslant \ell(c_2)$. This relation lets us extend this notion to the whole image, as captured in the following definition.

Definition 6.12. (First channel order) — *Given two images, $A_1, A_2 : \mathbb{Z}^n \to \mathfrak{C}$, we say that A_1 is (pixel-wise) lighter or equal than A_2, denoted as $A_1 \subseteq_\ell A_2$ if,*

$$A_1 \subseteq_\ell A_2 \iff \text{for all } x \in \mathbb{Z}^n, \ A_1(x) \leqslant_\ell A_2(x).$$

We remark that, although we introduce now an h-order, our mathematical morphology operators are not based on an h-order, but on the total order presented in Def. 6.1. This h-order is introduced in relation with two properties: the first channel monotonicity and the first channel adjunction.

Proposition 6.13. — *The relation $\leqslant_\ell$ for colors is a total preorder, and the relation $\subseteq_\ell$ for images is a preorder.*

Proof. Let us consider three colors $c_1, c_2, c_3 \in \mathfrak{C}$, and their first channels, $l_i = \ell(c_i)$ for $i \in \{1, 2, 3\}$. Then, either $l_1 \leqslant l_2$ or $l_2 \leqslant l_1$, so $\leqslant_\ell$ is a total relation—and, therefore, also a reflexive one. Also, if $l_1 \leqslant l_2$ and $l_2 \leqslant l_3$, then $l_1 \leqslant l_3$, which implies that $\leqslant_\ell$ is transitive. Thus, it is a total preorder.

With regard to the relation $\subseteq_\ell$, let us consider three images $A_1, A_2, A_3 : \mathbb{Z}^n \to \mathfrak{C}$. First, since $\leqslant_\ell$ is reflexive, for all $x \in \mathbb{Z}^n$, $A_1(x) \subseteq_\ell A_1(x)$, which implies that $\subseteq_\ell$ is a reflexive relation. Second, if $A_1 \subseteq_\ell A_2$ and $A_2 \subseteq_\ell A_3$, then for all $x \in \mathbb{Z}^n$, $A_1(x) \leqslant_\ell A_2(x)$ and $A_2(x) \leqslant_\ell A_3(x)$. Thus, $A_1 \subseteq_\ell A_3$. All of the above implies that $\subseteq_\ell$ is a preorder. $\square$

We remark that such relations are not partial orders, since antisymmetry does not hold. We can now state the monotonicity of our operators as follows:

Proposition 6.14. (Monotonicity) — *Let $A : \mathbb{Z}^n \to \mathfrak{C}$ be an image and let $B : \mathbb{Z}^n \to [0,1]$ be a structuring element such that $B(0) = 1$. Let C be a conjunction and I a fuzzy implication function. Then,*

$$\mathcal{E}_I(A, B) \subseteq_\ell A \subseteq_\ell \mathcal{D}_C(A, B). \tag{6.6}$$

Besides, if I is the R-implication of C, then,

$$\mathcal{E}_I(A, B) \subseteq_\ell \mathcal{O}_{C,I}(A, B) \subseteq_\ell A \subseteq_\ell \mathcal{C}_{C,I}(A, B) \subseteq_\ell \mathcal{D}_C(A, B). \tag{6.7}$$

Proof. Since our operators generalize the ones of the fuzzy mathematical morphology (Prop. 6.10) and the order only takes into account the first component (Def. 6.12), we can translate the result in [129, Prop. 45] to our framework. $\square$

6.3.5 First Channel Adjunction

The fuzzy mathematical morphology, which we extend, has the property of being an adjunction. That is, under certain constraints on the conjunction C and the fuzzy implication function I, $A_1 \subseteq \mathcal{E}_I(A_2, B)$ if and only if $\mathcal{D}_C(A_1, \bar{B}) \subseteq A_2$ (see Prop. 2.20). We remark that we use the reflection of the structuring element, $\bar{B}$, given our specific definition of erosion and dilation, similar to the definition by Kerre *et al.* [129, Def. 12].

Although we consider that the inclusion is not well defined between multivariate images, we can transfer this property to our setting if we restrict the inclusion to the first channel as an *h*-adjunction [76]. Thus, considering the first channel inclusion above defined, $\subseteq_\ell$, we may state the following property.

Proposition 6.15. (*h*-Adjunction) — *Let C be a left-continuous t-norm, I its residual implication and $B : \mathbb{Z}^n \to [0,1]$ a structuring element. Then, the operators $\mathcal{E}_I(\cdot, B)$ and $\mathcal{D}_C(\cdot, \bar{B})$ form an ℓ-adjunction. That is, for all images $A_1, A_2 : \mathbb{Z}^n \to \mathfrak{C}$,*

$$A_1 \subseteq_\ell \mathcal{E}_I(A_2, B) \iff \mathcal{D}_C(A_1, \bar{B}) \subseteq_\ell A_2.$$

Proof. Again we use the fact that our operators generalize the ones of the fuzzy mathematical morphology (Prop. 6.10) and the order restricted to the first component matches that of the fuzzy mathematical morphology. Therefore, we can translate the result in [129, Prop. 52] to our framework. $\square$

6.4 Comparison of Color Morphology Approaches

In this section, the Soft Color Morphology operators are compared with the morphological operators of other frameworks that deal with color images. They are compared both in terms of their

characteristics and with visual examples. In particular, Table 6.1 collects practical information regarding the nature and application of each algorithm: whether they preserve colors, their computational complexity, and so on. Later on, the visual comparisons are meant to provide an insight of their behaviour and compare it with the other frameworks. Since we are designing general operators, our aim is to make them as interpretable as possible, so practitioners know when and how to apply them. We remark that a quantitative comparison is not possible: performance metrics can only be used when facing a specific application, which is out of the scope of this chapter.

TABLE 6.1: Comparison of different approaches to color mathematical morphology. The superscript * indicates that the order is adaptive—that is, it depends on the image being processed, whereas the superscript ** indicates it depends not only on the color, but also on the spatial location of the pixel. The superscript † indicates that the generalization is valid only if properly customized.

		Type of order	Color space for color images	Computational time	Color preservation	Chromatic preservation	Generalization of mathem. morphology	Non-binary structuring elements	Customizable
1996	Component-wise [78]	Marginal	RGB	⧖	✗	✗	Binary [157]	✗	✗
1998	Bit mixing [47]	Reduced	RGB	⧖	✓	✓	Grayscale [75]	✗	✗
2001	Color reference [153]	Reduced	CIELab	⧖	✓	✓	Grayscale [75]†	✗	✓
2002	Lexic. order (VSH) [116]	Lexic.	HSV	⧖	✓	✓	Grayscale [75]	✗	✗
2012	Histogram-based [28]	Partial*	CIELab	⧖⧖⧖	✓	✓	None	✗	✓
2012	Random projections [182]	Reduced*	RGB	⧖⧖	✓	✓	None	✗	✗
2017	Col. ref. (CIELab Quantale) [178]	Reduced	CIELab	⧖	✓	✓	Grayscale [75]†	✓	✓
2017	Soft color morphology	Lexic.**	CIELab	⧖⧖	✓†	✓	Fuzzy [55]	✓	✓

During the rest of the section, we consider all the morphological frameworks contained in Table 6.1. Among other features, it explains the type of order between colors, as previously introduced in Sect. 6.1. The complexity of all of them is at least as high as computing the lattice-based definition of morphological operators, but it is higher in some situations: the histogram-based version by Benavent *et al.* [28] requires computing a histogram of size $100 \times 100 \times 100$ and filter it (or precomputing it to process images that are similar to a certain database); the random projections [182] technique requires approximating a supremum over an infinite set of projections with a maximum over a finite random projections uniformly distributed (specifically, 1000 projections are used in the original paper). The color preservation, chromatic preservation in natural images and generalization of more elementary mathematical morphology frameworks are interpreted as in Properties 6.8, 6.9 and 6.10.

We emphasize that our method uses structuring elements that can be encoded as grayscale images, whereas the CIELab quantale approach [178] considers multivariate structuring elements. Lastly, the customization classification goes beyond changing the structuring element: the color reference [153] and the CIELab quantale [178] morphologies depend on a user-chosen color; the histogram-based morphology [28] admits coarser or finer smoothings of the histogram, and the soft color morphology depends on a fuzzy conjunction and fuzzy implication function that model its behaviour for non-binary structuring elements.

Figures 6.5-6.9 are representations of the same morphological operator for the different methods in Table 6.1. All methods have been executed with the values of the parameters recommended in the original articles, using white as the color reference for the method by Sartor [153] and black as the color reference on the CIELab quantale [178]. The soft color morphology method uses the minimum t-norm as conjunction and its R-implication, the Gödel implication. To compare a method that admits non-binary structuring elements with another one limited to binary structuring elements, whenever a non-binary structuring element is considered, it is thresholded at the level 0.5 to be used by the latter. On the other hand, for the method by Valle *et al.* [178], grayscale structuring elements are mapped into the CIELab, based on the unity element. Lastly, all methods that employed an
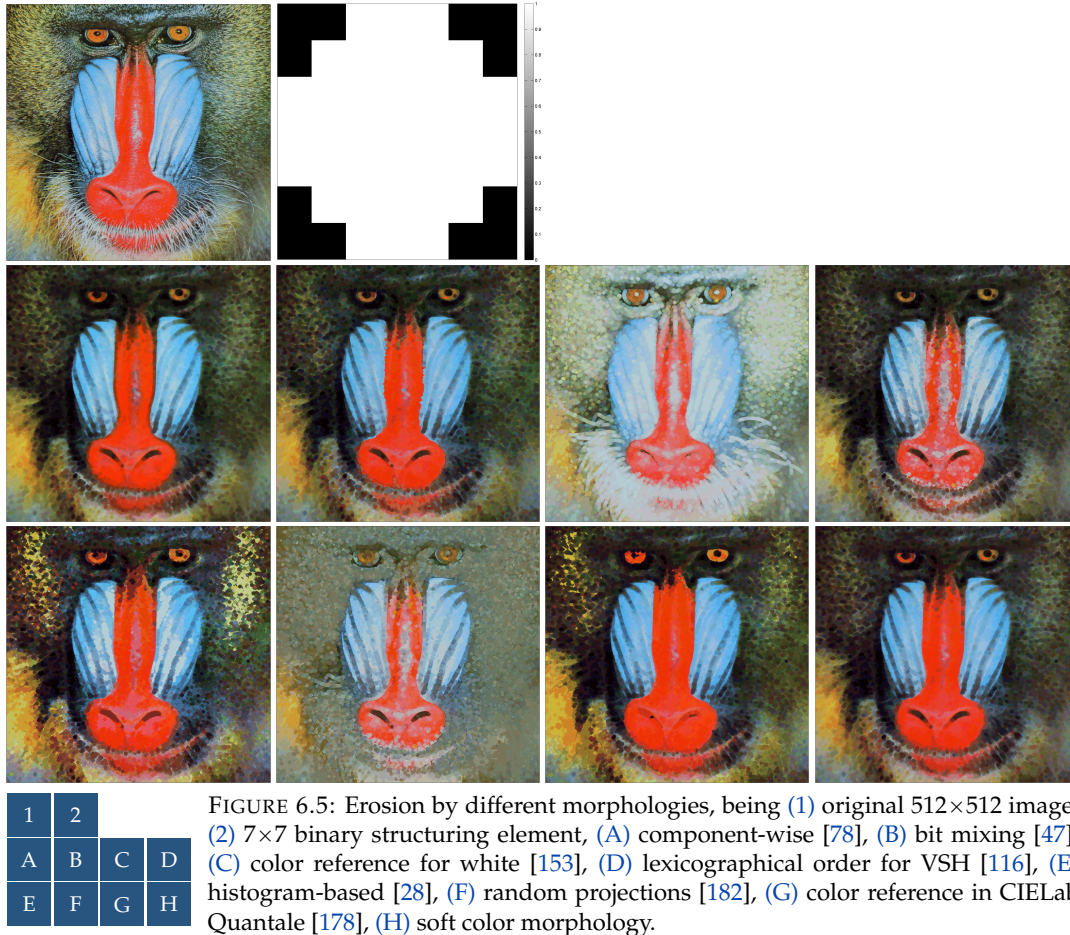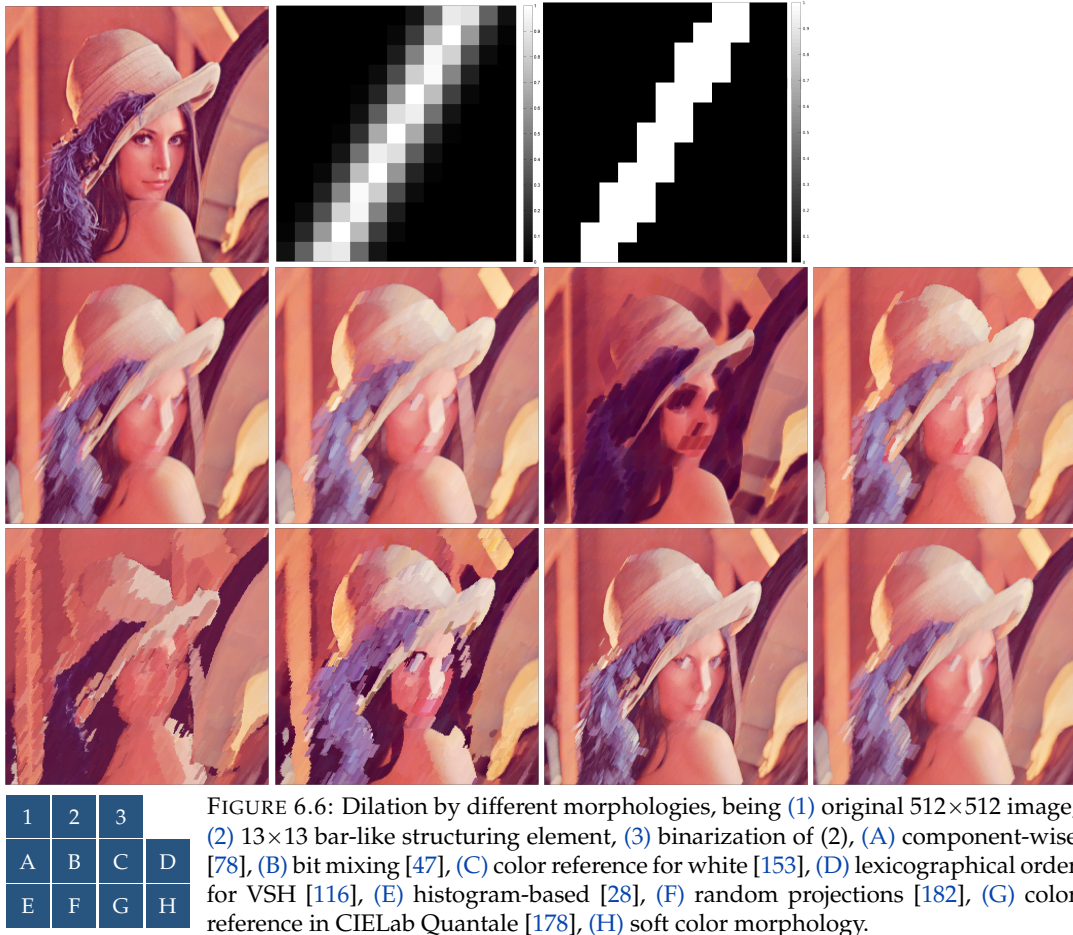
FIGURE 6.5: Erosion by different morphologies, being (1) original 512×512 image, (2) 7×7 binary structuring element, (A) component-wise [78], (B) bit mixing [47], (C) color reference for white [153], (D) lexicographical order for VSH [116], (E) histogram-based [28], (F) random projections [182], (G) color reference in CIELab Quantale [178], (H) soft color morphology.

adaptive ordering, did so the first time the original image was processed (not after each erosion or dilation).

Figures 6.5 and 6.6 depict the basic operators: erosion and dilation. We remind that erosion is meant to shrink objects, whereas dilation enlarges them. Color reference for white seems to work in an opposite direction as the other frameworks, whereas random projections, being image dependent, works counterintuitively in the first case, but intuitively in the second. It is noticeable how the histogram-based technique provides a sharp result in Fig. 6.5 and blurs the original image in Fig. 6.6, to the point of removing small objects, due to its image-dependent behaviour. The method based on lexicographic order for the channels VSH provides a singular result: the erosion contains large bright areas, such as the nose. Although the result of the component-wise operators seems similar to the soft color operators, we remark that the colors generated are different from those in the original image. This is not visually evident, but can be detected when comparing it pixel-wise with the original image. This is specially relevant when using other operators such as the top-hat or the morphological gradient. However, all of the techniques remove either the eyes or the eyelashes in Fig. 6.6, since they are almost perpendicular to the non-isotropic structuring element employed.

Figures 6.7 and 6.8 contain, respectively, the opening and the closing operators. These operators are meant to remove objects smaller than the structuring element and to remove holes between objects that are smaller than the structuring element. This behaviour makes them flatten heterogeneous regions, transforming textures into more uniform areas. In addition to it, the opening and the closing should keep the shapes of big objects.

In Fig. 6.7, we observe how the component-wise approach provides a visually similar image to the original one. Again, the image-dependent morphologies, in Fig. 6.7(e, f), provide significantly

1	2	3	
A	B	C	D
E	F	G	H

FIGURE 6.6: Dilation by different morphologies, being (1) original 512×512 image, (2) 13×13 bar-like structuring element, (3) binarization of (2), (A) component-wise [78], (B) bit mixing [47], (C) color reference for white [153], (D) lexicographical order for VSH [116], (E) histogram-based [28], (F) random projections [182], (G) color reference in CIELab Quantale [178], (H) soft color morphology.

different results, modifying the blue region the former and the red region the latter. We remark that only the soft color operators employ the non-binary structuring element, since it is the only paradigm that can meaninguflly handle it. The soft response provided by the non-binary structuring element provides both sharp contours and smooth regions.

With regard to the closing operator, in Fig. 6.8, most alternatives treat the white background as the object to be closed, except for the color reference and the random projections algorithms. The histogram-based method, as a matter of fact, is not able to deal with this artificial image and highly distorts the boundaries of the objects. Some approaches, most notably the histogram-based closing in Fig. 6.8(E), do not preserve the shape of the inner squares. The rest of them, including the soft color operators, are able to deal with this artificial image.

Fig. 6.9 represents a more complex morphological operator, the morphological gradient computed, which aims at locating the boundaries between objects. To compute it, we do so by considering the Euclidean distance between the CIELab-encoded dilation and the CIELab-encoded erosion for each of the frameworks studied (as in Def. 6.7). In this case, the methods with adaptive orderings— the histogram-based and the random projection ones—do not capture well the gradients near the mouth. The component-wise method with non-binary structuring element changes the value of pixels, and this has dramatic effects, since flat regions produce non-zero gradient values.

6.5   Analysis and Conclusions

In this chapter we formally presented the basic operators of the soft color mathematical morphology: the soft color erosion and the soft color dilation (Defs. 6.3 and 6.2), and we also combined
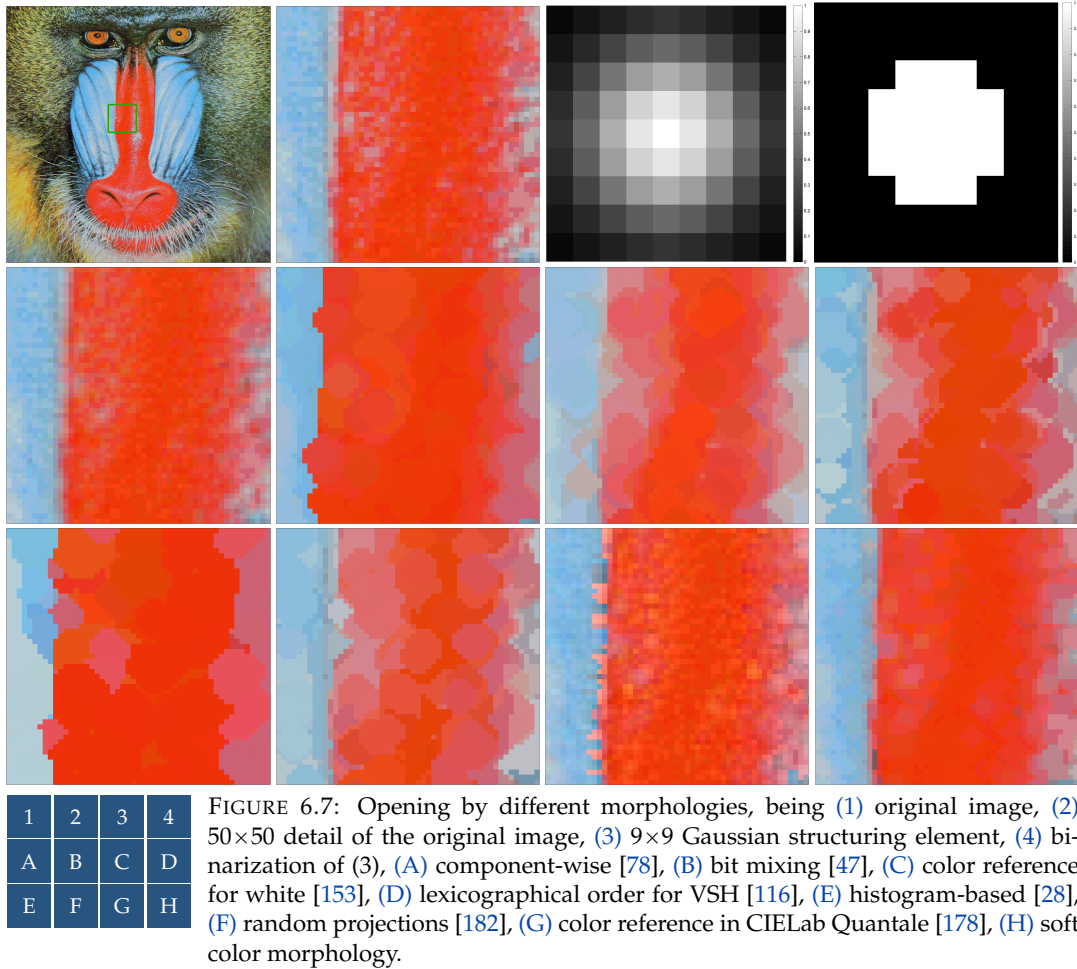
FIGURE 6.7: Opening by different morphologies, being (1) original image, (2) 50×50 detail of the original image, (3) 9×9 Gaussian structuring element, (4) binarization of (3), (A) component-wise [78], (B) bit mixing [47], (C) color reference for white [153], (D) lexicographical order for VSH [116], (E) histogram-based [28], (F) random projections [182], (G) color reference in CIELab Quantale [178], (H) soft color morphology.

them to create other morphological operators. Then, we studied their properties from a theoretical point of view in Sect. 6.3, and compared them extensively to other color morphologies in Sect. 6.4. We end the chapter with a discussion of the comparison and the strengths and weaknesses of the soft color morphology.

6.5.1 Analysis of the comparison

Regarding color preservation, the only morphology falling short is the component-wise morphology. It is not able to control whether new colors are created or not producing unexpected results when combining erosion and dilation. The rest of methods, which deal with vectors instead of independent channels, do not show this weakness.

With regard to interpretability, the soft color and the lexicographical order for VSH morphologies are interpretable: bright objects are enlarged with dilation and shrinked with erosion. This, however, is not always the best solution, as can be observed in Fig. 6.8: the closing operator is assumed to fill the holes within the objects. On the other hand, adaptive orderings fall short in interpretability: depending on the image, they may produce completely different results. For instance, the same object captured under two different backgrounds may be dealt with in a totally different way. Besides, extra care should be taken when processing artificial images, such as in Fig. 6.8, since they may alter unexpectedly the ordering.

Taking into account the visual appearance of results (that is, how *natural* the processed images look), almost all methods provide acceptable responses, even the most basic ones. With a closer
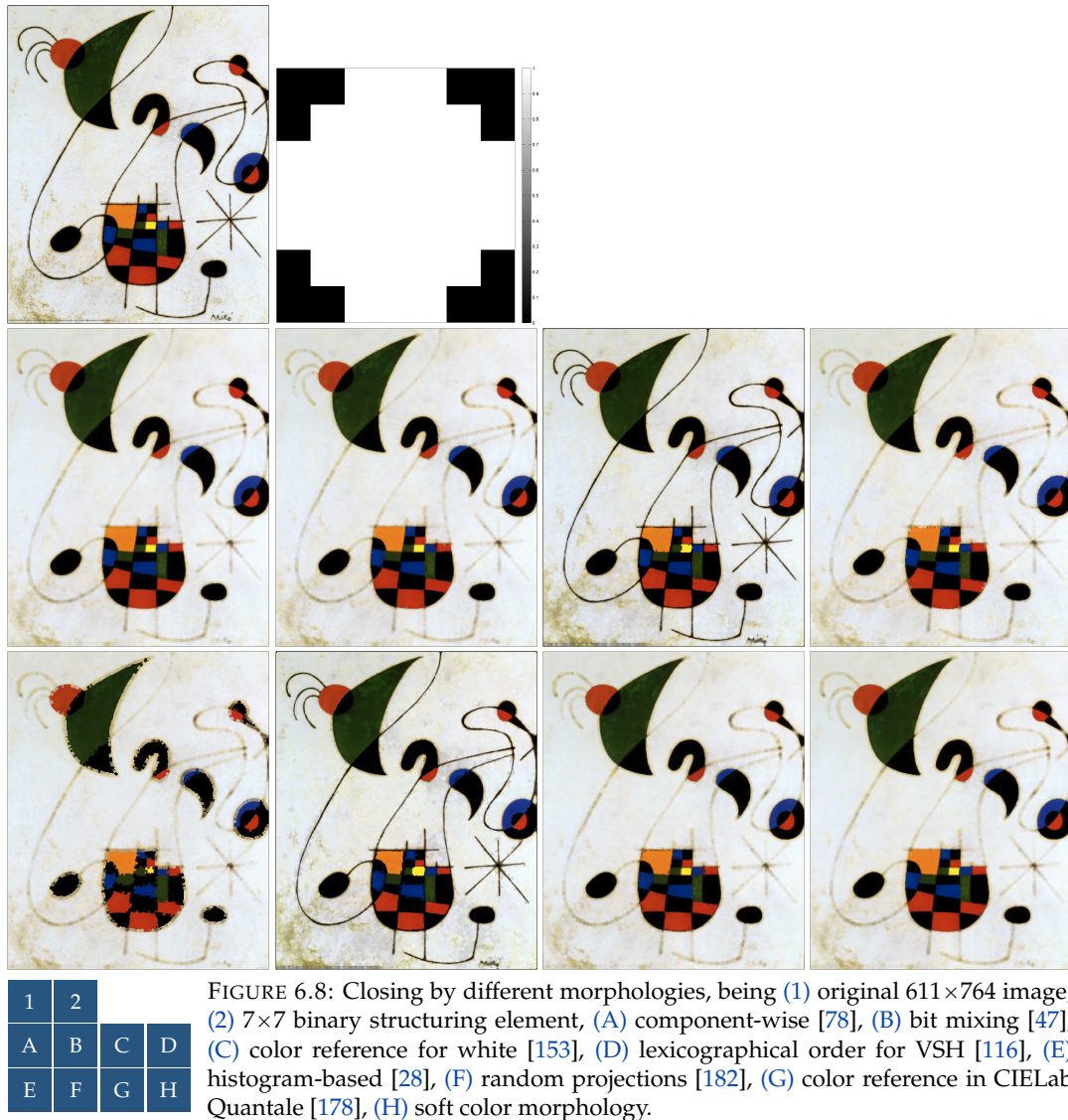
FIGURE 6.8: Closing by different morphologies, being (1) original 611×764 image, (2) 7×7 binary structuring element, (A) component-wise [78], (B) bit mixing [47], (C) color reference for white [153], (D) lexicographical order for VSH [116], (E) histogram-based [28], (F) random projections [182], (G) color reference in CIELab Quantale [178], (H) soft color morphology.

look, as in the texture-flattening operator in Fig. 6.7, the soft color morphology operators provide smoother outputs due to the possibility of using non-binary structuring elements.

Finally, the runtime of methods may also be important in some applications. All of them are appropriate for off-line image processing, although not for real-time image processing. The fastest methods are the ones based on a non-adaptive total ordering—such as morphologies based on bit mixing, color reference and lexicographical orders and the soft color morphology with binary structuring elements. These can leverage the lattice-based definition of erosion and dilation. The component-wise morphology consists on replicating this same operator once per channel, multiplying thus the computational complexity by a fixed constant. The soft color morphology operators do not replicate the same operator several times, but whenever a non-binary structuring element is used, we must evaluate the fuzzy conjunction (or fuzzy implication function) several times. Computationally, this is roughly equivalent to multiplying the computational complexity of the lattice-based definition by as many different values as the structuring element has. Lastly, the adaptive orderings studied here are the most time-consuming methods. They spend the majority of time creating the adaptive ordering, which has to be done once per image. The histogram-based method proposes a solution to this problem, consisting on reusing the same histogram if the processed image is similar to a previously computed one from a statistical point of view.
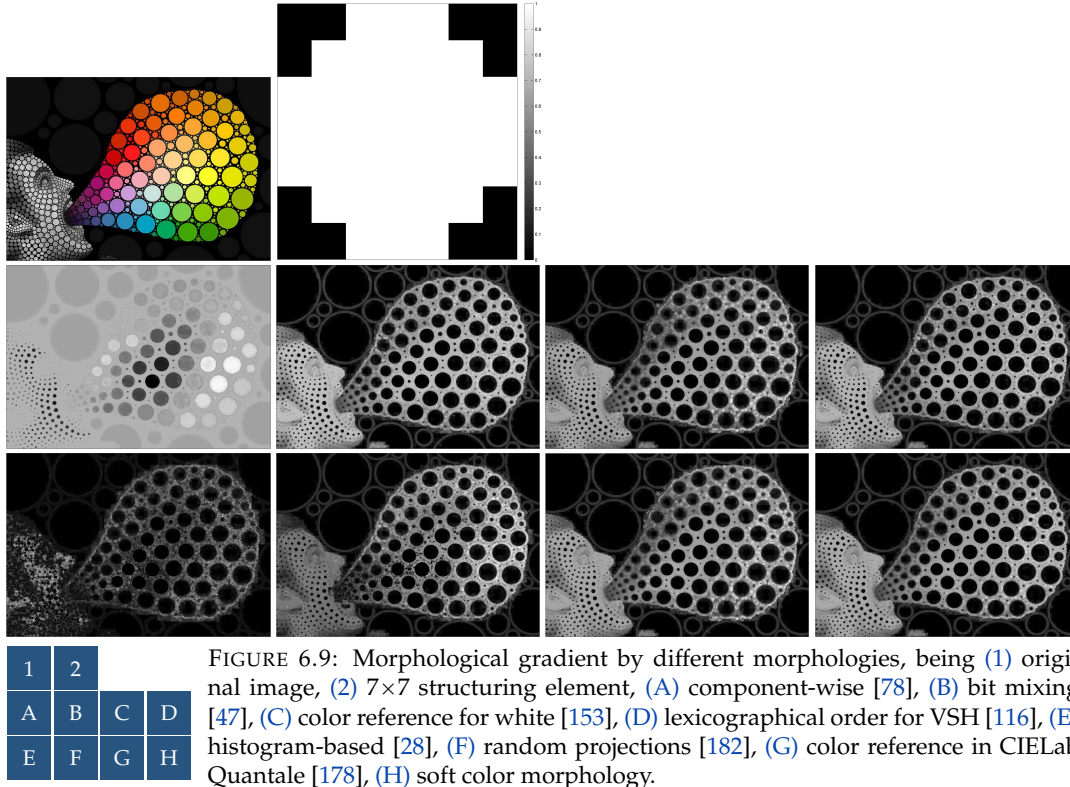
FIGURE 6.9: Morphological gradient by different morphologies, being (1) original image, (2) 7×7 structuring element, (A) component-wise [78], (B) bit mixing [47], (C) color reference for white [153], (D) lexicographical order for VSH [116], (E) histogram-based [28], (F) random projections [182], (G) color reference in CIELab Quantale [178], (H) soft color morphology.

6.5.2 Strengths and Weaknesses of the Soft Color Morphology Operators

The interpretability of the operators is of paramount importance. Mathematical morphology encompasses operators that manipulate the shape of objects, and thus they must implicitly define what is *object* and what is *background* in an image. This distinction is harder in color or multivariate images than in grayscale or binary images. Our approach is to consider that pixels that belong to the background have higher values in the first channel. This is very intuitive for color images encoded with the CIELab color space.

On the other hand, these operators can also process images encoded in other color spaces and general multivariate images—like hyperspectral imagery with a large number of channels. However, in this case the practitioner must choose the channel that best distinguishes the objects of interest, or estimate this information from a series of channels (for instance, with the histogram-based approach by Benavent *et al.* [28]).

The soft color morphology operators fulfill several interesting theoretical properties. Firstly, they preserve the chromatic information of pixels when using the CIELab color space for color images. Secondly, they can be tuned to preserve full colors, simply by using a binary structuring element. We remark that creating new colors may be desirable in some situations like morphological filtering, where smoothness is favoured over realism, although it may be unacceptable in other application fields. If color preservation is pursued, the soft color erosion and dilation become equivalent to the lattice-based definition of dilation and erosion, with an order that is highly interpretable to deal with color images. Thirdly, these operators generalize the fuzzy mathematical morphology ones, which ensures the monotonicity/antimonotonicity of erosion and dilation, among other properties. This is also responsible for the soft outputs achieved with non-binary structuring elements. However, the soft color closing and opening do not inherit the idempotence from the fuzzy mathematical morphology operators. This, which is due to possible ties when finding a *maximum* or *minimum* color (see Def. 6.1), is not a big problem in practice since ties are infrequent in natural images.

The computational complexity is comparable to that of the fuzzy mathematical morphology for grayscale images, which is low but still greater than that of the lattice-based definition of mathematical morphology. On a CPU Intel$^{©}$ Core^{TM} i5 @ 3.10 GHz, the time to process one soft color dilation or soft color erosion ranges from 81 ms (256×256-pixel images) to 1.56 s (1024×1024-pixel images) with MATLAB$^{®}$ code, using the minimum conjunction, the Gödel implication and a 5×5-pixel Gaussian-shaped structuring element. This performance could be greatly improved with parallelization and GPU-optimized code, further achieving performances enough to be embedded into real-time systems, as can be seen in Ch. 5 in which an equivalent complex operator achieves such a real-time performance.

Chapter 7

Hair Removal in Dermoscopic Images

Dermoscopic images provide useful information towards the diagnosis and classification of skin lesions, such as dysplastic nevi, lentigo maligna or nodular melanomas, among others. Several computer vision algorithms have been developed to do so automatically, based on expert knowledge. A first preprocessing step may be cleaning the image to be able to segment the lesion. In this chapter, we introduce an algorithm to reduce the noise in dermoscopic images. This includes the bubbles caused by the immersion fluid and, foremost, the skin hair that is occasionally present. We remark that hairs in dermoscopic images can be interpreted as dark curvilinear objects according to the definition given in Ch. 3. Thus, they can be detected with a similar strategy to the ones used in Ch. 4.

In Sect. 7.1 we study the specifics of such task, such as typical image settings, and the related literature. Sects. 7.2 and 7.3 introduce the building blocks for the hair removal algorithm, presented in Sect. 7.4. We remark that the core of our proposed solution is exclusively based on the soft color morphological operators introduced in Ch. 6. We conclude this chapter comparing this method with other state-of-the-art algorithms in Sect. 7.5.

7.1 Introduction

Malignant melanoma, or simply melanoma, is the most dangerous form of skin cancer. Although it only represents 4% of all skin cancers, it causes the 80% of deaths related to skin cancer [86] and presents peaks of appearance between 20 and 45 years of age [88]. In Spain, approximately 5000 new patients are diagnosed each year, which is a quantity increasing around 7% yearly [86].

Dermoscopic imagery is a non-invasive and effective tool towards the clinical diagnosis of skin lesions [115], both for superficial spreading ones but also in the case of vertically growing melanomas [8]. Such images are acquired with a dermatoscope, which is a camera specially designed to capture small regions and to avoid the reflection of light on the skin surface. Besides employing with appropriate optics, dermatoscopes avoid such reflections with polarized or non-polarized light and an immersion fluid (*e.g.* alcohols or mineral oil).

Computer-aided image analysis technologies can be leveraged to design diagnosis support tools and thus help practitioners. The impact of such tools is expected to increase due to the growing quality of dermoscopic devices along with their low cost and high availability. There are, however, several difficulties when dealing with dermoscopic images. To automatically study the objects they contain, one of the first steps is the reduction of the noise they present, which includes bubbles caused by the immersion fluid and skin hair. In this chapter, we present a procedure that removes hair from dermoscopic images and also reduces the amount of artifacts caused by the immersion fluid, while largely preserving the texture of skin lesions.

A variety of techniques have been proposed to remove hair from dermoscopic images. The methods that effectively remove hair in the literature present a two-step pattern: segmentation and inpainting. The first step, segmentation, is aimed at creating a binary mask tagging the pixels that will be replaced in the second one, inpainting, with colors that mimic the original skin area.

A first important algorithm is Dullrazor®, proposed by Lee *et al.* in [106]. It uses single-channel top-hat transforms by closing with three different orientations, followed by a pre-specified threshold, to identify hair. Three binary masks are obtained, one per each of the RGB channels, and the union of such masks is computed and further refined to avoid objects whose shape is significatively different to a curvilinear object. Hair pixels are replaced by bilinear interpolation between two close pixels, and further smoothed by an adaptive median filter. On the other hand, Xie *et al.* [192] introduced a method that also employs grayscale morphological top-hat transforms followed by a refinement based on a specific elongation measure. Their inpainting method is based on partial differential equations. Inpainting is done with non-linear diffusion filters that preserve the contours of the lesion by propagating linear structures. Nguyen *et al.* [132] employed matched filters to face the same task. Their matched filtering approach uses 18 bar-like kernels with a Gaussian decayment profile. The maximum response is computed out of 18 rotated kernels. This output is further processed by filtering, binarization and skeleton extraction. The centerlines represented by the skeleton are fitted to a Gaussian cross-section profile to avoid false positives and include previously rejected positives. Similarly to [106], the inpainting step consists on a linear interpolation based on the neighbouring non-hair pixels. Abbas *et al.* [1] used Derivative of Gaussians (DoG) as the kernels of linear filters to extract the curvilinear structures that hairs represent. After some morphological processing including region linking, circular region removal and hole-filling, the selected pixels are inpainted with a fast-marching based algorithm. It is a non-iterative algorithm that transports the image values in a coherence direction, which is obtained as the eigenvector of minimal eigenvalue of a structure tensor. Better results are achieved when the color space used is CIELab instead of RGB. Kiani and Sharafat [91] used the Radon transform in grayscale dermoscopic images to detect the local orientation, which is used as the orientation of the Prewitt edge detector. The removed pixels are replaced with the average background value and several smoothing averages are performed to create the final result. Toossi *et al.* [171] presented a method that detects hairs by means of a modified Canny edge detector and a morphological post-processing to dilate, remove blobs and fill holes in the binary mask. Their inpainting procedure combines coherence transport inpainting—as in [1]—with wavelet decomposition.

Table 7.1 contains a summary of the techniques classified according to their segmentation and inpainting strategy.

TABLE 7.1: Detection and inpainting techniques employed in the literature to remove hair from dermoscopic images. The superscripts indicate whether the procedures use interchannel information, denoting $\rightarrow$ that only one channel is used; $\equiv$ when all channels are processed independently; and $\Rightarrow$ if all channels are processed simultaneously.

Year	Paper	Hair segmentation	Inpainting method	Color space
1997	Lee *et al.* [106]	Grayscale top-hat$^\equiv$	Bilinear interpolation$^\equiv$	RGB
2009	Xie *et al.* [192]	Grayscale top-hat$^\equiv$	Non-linear PDE$^\equiv$	RGB
2010	Nguyen *et al.* [132]	Matched filtering$^\rightarrow$	Linear interpolation$^\equiv$	RGB
2011	Abbas *et al.* [1]	Derivative of Gaussians$^\rightarrow$	Coherence transport$^\Rightarrow$	CIELab
2011	Kiani *et al.* [91]	Prewitt edge detector$^\rightarrow$	Iterative smoothing$^\rightarrow$	Grayscale
2013	Toossi *et al.* [171]	Canny edge detector$^\rightarrow$	Coherence transport$^\Rightarrow$	RGB
	This work	Color top-hat$^\Rightarrow$	Morph. inpainting$^\Rightarrow$	CIELab

Our goal is to propose and study a morphological algorithm to remove hair in color dermoscopic imagery. Both the segmentation of hair and the inpainting algorithms are designed taking into account the nature of dermoscopic images: the presence of other sources of noise (*i.e.* bubbles), the limited field of view and the need of maintaining the texture and other characteristics of the lesion to further study it. We design both the segmentation and inpainting stages using the soft color morphology operators from Chapter 6.

7.2 Curvilinear Object Detector with Soft Color Morphology Operators

In this section, we introduce a curvilinear object detector for color images based on the soft color morphology from Chapter 6. More specifically, it is a combination of soft color top-hat transforms (see Def. 6.6, page 87).

The detector introduced in the following is based on the curvilinear detector from Sect. 4.3. There are, however, some differences that must be emphasized. First, this detector is designed to process color images, rather than grayscale ones. It must, therefore, use CIELab-encoded images. Second, we consider the maximum of several outputs minus their minimum (compare Fig. 4.7, page 56 with Fig. 7.1, page 101). This operation is intended to discriminate isolated pixels. Taking into account the task at hand, not considering the effect of isolated pixels would have a non-despreciable impact on the final result.

Given the curvilinear appearance of hair in dermoscopic images, we observe that edge detectors are not suited for this task. Although they may seem similar algorithms, we are not interested in locating the boundary between two objects. Instead, our aim is to detect a thin, elongated object placed in front of a somewhat uniform background. For more details on this difference, the reader is referred to Sections 3.1 and 3.2.

The curvilinear object detector is based on a combination of soft color black top-hat operators (see Def. 6.6, page 87). As previously discussed in Sect. 6.2, the black top-hat, or top-hat by closing, highlights the thin and dark regions that have completely disappeared during the dilation operation. The formal definition of our detector $\mathcal{D}$, visually depicted in Fig. 7.1, is:

$$\mathcal{D}(A) = \max_{\alpha} \left\{ \mathcal{BTH}_{C,I}(A, B_{\alpha}) \right\} - \min_{\alpha} \left\{ \mathcal{BTH}_{C,I}(A, B_{\alpha}) \right\} \tag{7.1}$$

where C is a conjunction, I a fuzzy implication function and α ranges over different orientations of the structuring element B_{α}. The different top-hat operators differ only on the orientation of the structuring element B. All of the structuring elements B_{α} are bar-like shapes with a Gaussian decayment enclosed in a squared region. We remark that the maximum of the transforms outputs a high value on both isolated dark pixels and dark thin regions, whereas the minimum only does so in isolated dark pixels.
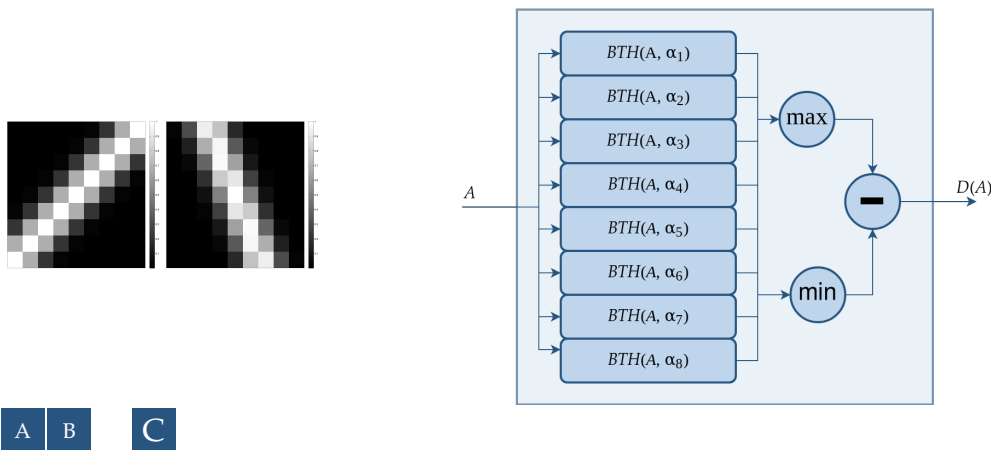

FIGURE 7.1: Two examples of structuring elements, $B_{45°}$ (A) and $B_{112.5°}$ (B), and workflow of the curvilinear object detector (C).

We leverage the appearance of hairs in dermoscopic images by fine-tuning the orientations and size of the structuring elements. Hairs appear as thin, elongated regions with clearly differentiated photometric features than those of its local background, which is non-uniform. More specifically, they are darker than the background, with possible brighter surroundings due to noise in acquisition and to lossy image formats. Besides, their width usually ranges between 4 and 9 pixels and the

ratio of hair pixels can range from 0% to almost 40% in the images we consider [126]. Specifically, the parameters used are the minimum t-norm as the conjunction $C(x,y) = T_{\mathbf{M}}(x,y) = \min(x,y)$; and its residual implication, the Gödel implication $I(x,y) = I_{\mathbf{GD}}(x,y)$ (See Sect. 2.3). We consider structuring elements at 8 different orientations enclosed in 9×9 regions. We remark that these parameters have been experimentally set.

7.3   Inpainting with Soft Color Morphology Operators

In this section, we present an inpainting method for color images based on the soft color morphology operators (see Sect. 6.2), which is inspired in the grayscale salt and pepper noise filter presented in [72]. The algorithm iteratively applies a morphological operator to recover missing pixels based on close ones, while maintaining non-missing pixels.

Inpainting is an image processing technique that consists on recovering the values of missing pixels [73, 156]. That is, considering that some pixels have a corresponding color associated, whereas others have not, inpainting the latter ones is assigning them a color so that the resulting image is visually plausible. Inpainting is useful in a variety of situations: to filter noise, to recover printed photographies that have been physically damaged or to design augmented-reality.

In each location $x \in \mathbb{Z}^2$, the original image contains either a color (denoted as $A(x) = c$, which is an element of the color space, $c \in \mathscr{C}_1 \times \ldots \times \mathscr{C}_n$) or a special symbol, denoting that this specific value is missing (denoted as $A(x) = \bot$ or, alternatively, stating that $A(x)$ is *missing*). In this context, the dilation and erosion from Defs. 6.3 and 6.2 are slightly modified: (i) pixels that are *missing* in the original image are ignored when finding the maximum and minimum, and (ii) whenever there is no candidate available for such maximum or minimum, the result asigned to the current location is the value *missing*: $\bot$.

Before presenting the inpainting algorithm, we introduce a morphological operator, $\mathcal{R}$, that recovers *missing* pixels based on nearby ones that are not *missing*. Whenever a *missing* pixel is overwritten, it is tagged as recovered and it is assigned the value provided by the operator. It is defined as the arithmetic mean of the closing and the opening, where the underlying erosions and dilations are modified as explained above. More formally, given an image A_n, we compute the *Recovery* operator, $\mathcal{R}(A_n)$, as:

$$\mathcal{R}(A_n) = \frac{\mathcal{O}_{T_{\mathbf{M}}, I_{\mathbf{GD}}}(A_n, B_5) + \mathcal{C}_{T_{\mathbf{M}}, I_{\mathbf{GD}}}(A_n, B_5)}{2}$$

where B_5 is a 5×5, flat, rounded structuring element, and where the average of a *missing* value, $\bot$, and any other value is defined as *missing* $\bot$. We remark that the combination of closing and opening is intended to avoid amplifying just bright or dark objects, while retaining the shape of objects.

We can now define the inpainting of an image A. The algorithm rewrites the pixels that are still *missing* as soon as enough information is available, while always maintains pixels whose value is already known. More formally, the iterative inpainting of an image A is defined as the limiting case of the following series:

$$A_1 = A, \quad \forall n \in \mathbb{N}, \ A_{n+1}(x) = \begin{cases} A_n(x), & \text{if } A_n(x) \neq \bot, \\ \mathcal{R}(A_n)(x), & \text{if } A_n(x) = \bot \text{ and } \mathcal{R}(A_n)(x) \neq \bot, \\ \bot, & \text{otherwise.} \end{cases}$$

The iterative inpainting of an image sucessfully recovers uniform or thin regions. Flat regions with missing pixels are perfectly recovered. Gradient-like textures or thin missing regions tend to be recovered successfully given the appearance of close, similar non-missing pixels in natural images. Besides, two uniform regions with missing pixels in the boundary are recovered with a small and soft transition. On the other hand, this inpainting procedure is not appropriate to fill

large and rounded missing regions, since complex textures are not recovered. Given the reduced thickness of hair, this does not pose a problem in our context.

Fig. 7.2 contains a visual example in which 90% of all pixels have been randomly tagged as missing. The rest 10% of pixels are iteratively processed in order to recover the whole image in the third iteration, where the shape of the objects and their photometric features are successfully recovered. We remark that a low-resolution image (256×256 pixels) has been used to help visualize the impact and shortcomings of the method.
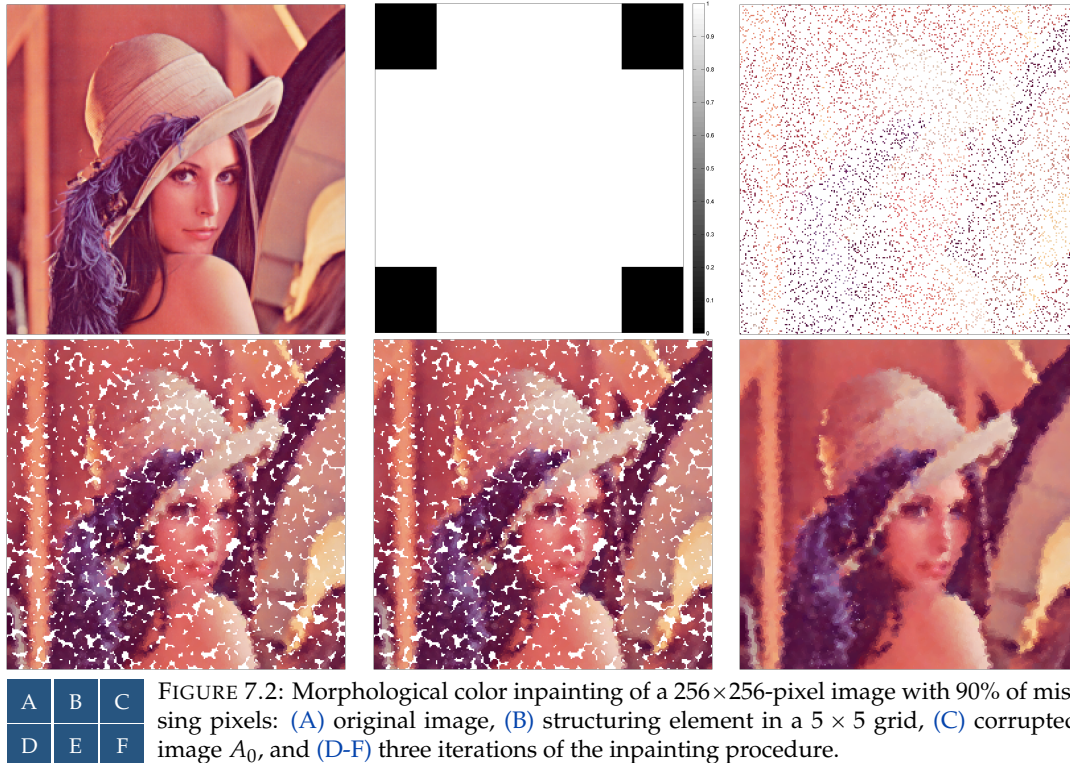

A	B	C
D	E	F

FIGURE 7.2: Morphological color inpainting of a 256×256-pixel image with 90% of missing pixels: (A) original image, (B) structuring element in a 5 × 5 grid, (C) corrupted image A_0, and (D-F) three iterations of the inpainting procedure.

7.4 Hair Removal Algorithm

Once the two main building blocks have been introduced, the full hair removal algorithm is presented. The course of the algorithm, composed by a series of sequential operations, is presented visually in Fig. 7.3 and, more formally, as follows.

1. The original image is encoded according to the CIELab color space (see Fig. 7.3(A)). We emphasize that we divide all values by 100 so that the first channel, L*, is exactly $[0, 1]$ (see Sect. 2.2).

2. The image is preprocessed with the Contrast Limited Adaptive Histogram Equalization—or CLAHE (see Sect. 2.5.2. In particular, the CLAHE is applied to the lightness channel, L*, while the chromatic channels a* and b* are preserved. (see Fig. 7.3(B)).

3. The preprocessed image is used as input to the curvilinear object detector introduced in Sect. 7.2 (see Fig. 7.3(C)).

4. The output is filtered with a 9×9 median filter to avoid large regions (see Fig. 7.3(D)) and it is thresholded (with threshold $t = 0.1$ when working with CIELab images scaled by a factor of 100) to create a binary mask. This binary mask is enlarged two pixels in each direction.
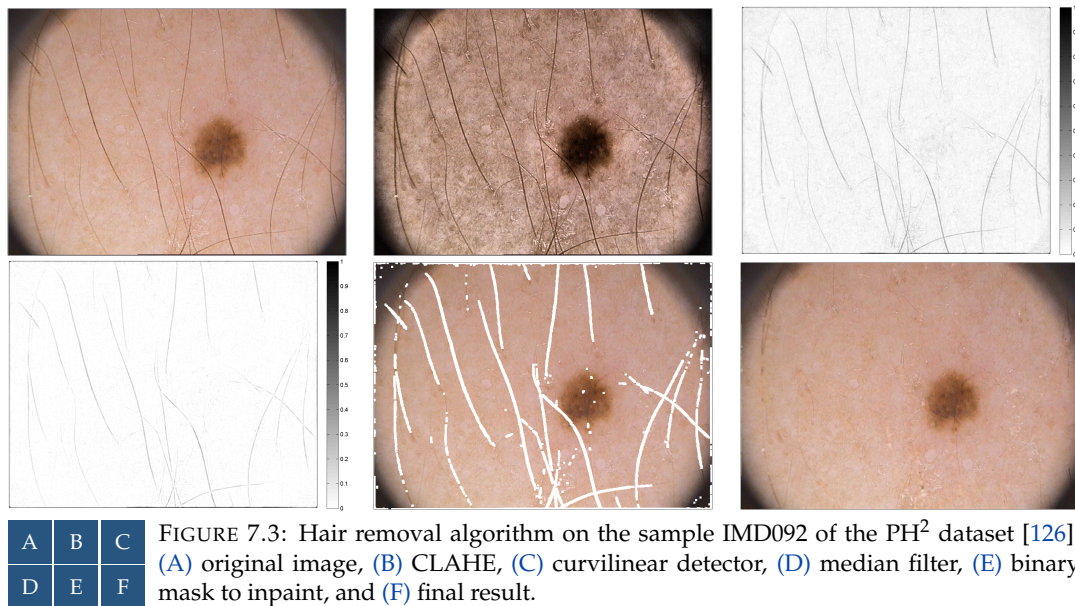
FIGURE 7.3: Hair removal algorithm on the sample IMD092 of the PH^2 dataset [126]. (A) original image, (B) CLAHE, (C) curvilinear detector, (D) median filter, (E) binary mask to inpaint, and (F) final result.

5. The pixels in the binary mask are removed from the original image and are tagged as *missing* (see Fig. 7.3(E)).

6. Lastly, such missing pixels are assigned a color value according to the inpainting procedure from Sect. 7.3, which provides the final result (see Fig. 7.3(F)).

It is designed for middle-sized dermoscopic images (approximately 600×600 pixels) with different pathologies, like the ones in the PH^2 dataset [126]. As can be observed in Fig. 7.3, our algorithm detects and inpaints almost all hair. Besides, it also succeeds at maintaining the features of the lesion.

7.5   Experiments and Conclusions

In this section, we compare our algorithm with two state-of-the-art methods, which have been selected due to their good performance.

Only visual comparison is feasible to compare the behaviour of different hair removal algorithms. This is due to the lack of ground truth and to the fact that a hair removal algorithm should maintain the lesion's features, specially its shape, color and texture, which can not be automatically evaluated. In Fig. 7.4 we show three original images (left), the result of the Dullrazor® algorithm [106] (center left), the result by Toossi *et al.* [171] (center right) and our algorithm (right).

As mentioned before, our algorithm effectively detects and inpaints almost all hair, while hairless images remain almost unchanged, and it maintains the features of the lesion. The shape and size are never affected due to the behaviour of the closing and opening when inpainting. The color is not affected either: colors with new chroma can not appear due to the nature of the soft color morphological operations. The texture of the lesion, on the other hand, is maintained in the majority of situations. In contrast with other procedures that employ smoothing filters or averages, our inpainting procedure does not tend to blur lesions: it recovers uniform regions correctly, and creates smooth but small transitions when inpainting the missing contours of different regions. The method by Toossi *et al.* leaves a considerable amount of hair and does not respect textures in general. Dullrazor®, on the other side, removes a fair amount of hair. However, it fails at recognizing hair within the lesion, and does not always preserve textures when it does so.
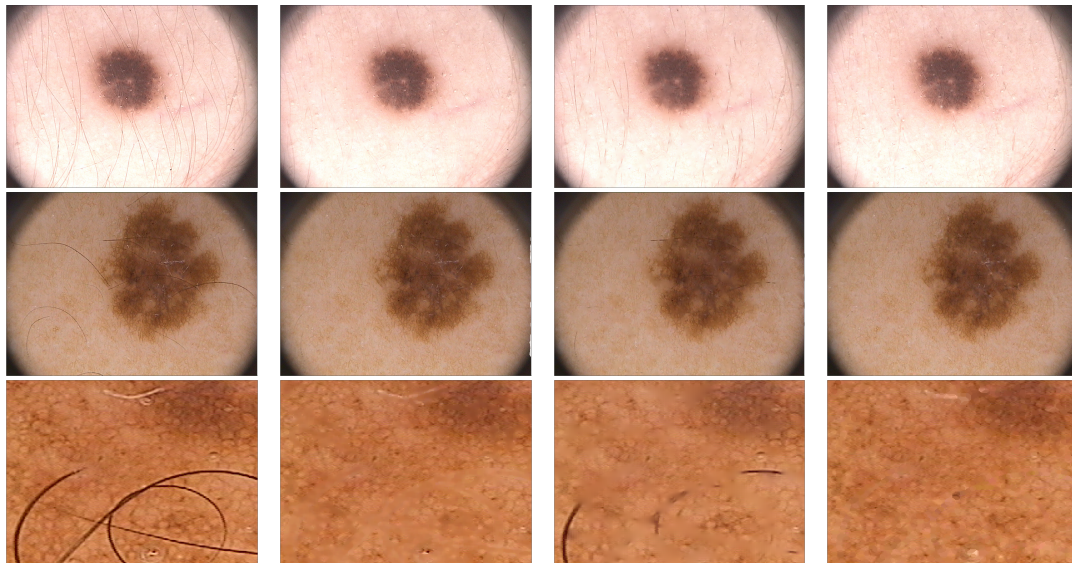
A1	A2	A3	A4
B1	B2	B3	B4
C1	C2	C3	C4

FIGURE 7.4: Comparison of hair removal algorithms. (1) Original, (2) Dullrazor®
by Lee *et al.* [106], (4) Toossi *et al.* [171] (3) and our algorithm; (A) sample IMD039,
(B) sample IMD047, and (C) a detail of sample IMD101 of the PH^2 dataset.

The shortcomings of our procedure are mostly found when the occluded region is too large (*i.e.*
exceeding 30-40% of the whole image) or with extremely wide hair, situation in which all the algo-
rithms compared tend to produce inaccurate results. Besides, as noted by Abbas [1], hair removal
algorithms tend to fail in distinguishing hair and curvilinear structures within the lesion such as st-
reaks, which alters the lesion texture. Although this effect has been minimized, it is still noticeable
in some images.

Future Work

The contributions in this thesis represent an extension of some lines of research that are, of course, not resoluted. First, it furthers the extent of some results, such as the accuracy of real-time vessel segmentation in retinal images. Therefore, it contributes towards extending these specific topics. Second, and not least, throughout this thesis some problems have not been fully addressed. They are not the main focus of the thesis, but are side problems or extensions that would be interesting to address due to different reasons. After developing the contributions in the previous chapters, we can collect several such problems that, to our knowledge, have become viable and are at the frontier of today's knowledge.

We enumerate the challenges arised throughout the previous chapters in the following.

- *Create a benchmark to compare curvilinear object segmentation algorithms.*
  A vast amount of methods have been developed to address a similar task (see Ch. 3). It is of paramount importance to study their relative effectiveness with the task they are supposed to face. To do so, we propose the creation of a well-tailored benchmark, associated to concrete performance measures, such as the ones in Sect. 3.4.2.

  An objective comparison between general-purpose curvilinear objects is not an easy task. It would be naive to think that all the methods studied have been designed taking into account the same curvilinear objects. Therefore, some of them will perform better in some situations, such as segmenting objects with very limited curvature, or processing high-noise images. An appropriate comparison should convey information about the performance of algorithms with regard to this variability. With this benchmark, practitioners would be able to select the method that best suits their needs.

- *Study combinations of soft color morphology operators.*
  It seems worth of study whether the soft color morphology operators are as versatile as their binary or grayscale counterparts. To do so, a first strategy is to combine the soft color erosion and the soft color dilation into more complex operators. In this thesis, we employ a color top-hat operator for pattern recognition (see Sect. 7.2) and a color inpainting algorithm (see Sect. 7.3). Besides extending the erosion and the dilation to a multivariate setting, we have studied the theoretical properties of the morphological operators that can be created with them. Also, we have also extended the notion of distance between two natural color images by using the Euclidean distance in the CIELab color space.

  Many more combinations have been proposed for binary or grayscale images, either applying them sequentially or iteratively. The hit-or-miss transformations poses as another example, for which the notion of *being included in a region* must be extended to a multivariate setting. Noise filtering can be implemented using sequences of opening and closing operations. Morphological contrast mapping is defined as iteratively replacing each pixel's color by the most similar color of the corresponding pixel between its dilation and its erosion. More combinations of erosion and dilation can be found in [75, Ch. 9].

- *Employ soft color morphology with n-dimensional and hyperspectral images.*
  We have studied the behaviour of the soft color morphological operators when dealing with natural color images (see Sections 6.4 and 7.4). However, these operators are designed to

process any type of color space (see Sect. 6.2). This line of work consists on studying the feasibility of employing mathematical morphology with new types of images.

For instance, we could design segmentation techniques for corregistered PET/RM images. Both PET and RM are single-channel, three-dimensional volumetric images. Thus, a corregistered PET/RM can be regarded as an image of the form $A : \mathbb{Z}^3 \to \mathbb{R} \times \mathbb{R}$. Although each modality is acquired based on a different underlying physical phenomena, they share the same spatial reference when they are corregistered. We could consider, for example, the task of segmenting a tumoral region provided its appearance in each of the imaging modalities. Also, it seems very interesting the application of soft color morphology to extract regions in kinetic PET images. In this case, they are images of the form $A : \mathbb{Z}^3 \times \mathbb{Z} \to \mathbb{R}$, having three physical dimensions and a discretized temporal axis. These images can be used to, for instance, capture the evolution of tissues while absorbing a certain radiotracer.

- *Analyse the impact of automatic hair removal in dermoscopic images.*
  Besides following the research line of further studying dermoscopic images, we are interested in analysing the hair removal step (see Ch. 7). More specifically, we contemplate studying how it improves the lesion segmentation and how it worsens the quality of the image. We emphasize that the inpainting algorithm is generating new information by filling the skin regions under the hair. The texture of the lesion is, thus, potentially distorted. This may pose as a very serious problem: the practitioner will potentially see spurious imprecisions, which could misguide his judgment. Studying this effect is an essential step towards validating it for a potential clinical use or to limit its results to be used by automatic algorithms.

- *Transfer computer science techniques to the medicine.*
  We recall that Objective 3 was aimed at bridging the gap between computer scientists and physicians. Given its broad nature, this objective has just been partially addressed in this thesis. Chapters 3 and 4 are aimed at reusing work between professionals from different fields. Ch. 7 was outlined taking into account the first outcomes of a collaboration with an expert dermatologist, Dr. del Pozo, from the department of the dermatology of the University Hospital *Son Espases*. Also, the author has been collaborating with Dr. Rubí and his team, from the nuclear medicine department of the same hospital, to model how high-grade cerebral tumors acquire a certain nutrient. Although this work in progress is not a constitutive part of the thesis, it has notably contributed to the learning process that has culminated in this thesis. All the above experiences have revealed the obstacles found when planning the deployment of computer science techniques into a medical environment.

This future line of work aims at creating, locally, an appropriate environment to design, create and deploy medical products. Such an environment requires a strong collaboration. Physicians perfectly know their needs, whereas other researchers are more aware of what is doable using the latest developments in their fields, and how to achieve it. We consider that deploying scientific discoveries is not a technical requirement, but an integral part of science.

Bibliography

[1] Q. Abbas, M. E. Celebi, and I. F. García. Hair removal methods: a comparative study for dermoscopy images. *Biomedical Signal Processing and Control* 6.(4) (2011), 395–404 (cited on pages 100, 105).

[2] M.-A. Abdul-Karim, K. Al-Kofahi, E. B. Brown, R. K. Jain, and B. Roysam. Automated tracing and change analysis of angiogenic vasculature from in vivo multiphoton confocal image time series. *Microvascular research* 66.(2) (2003), 113–125 (cited on pages 21, 33).

[3] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, tensor analysis, and applications*. Volume 75. Springer Science & Business Media, 2012 (cited on page 35).

[4] G. Agam, S. G. Armato, and C. Wu. Vessel tree reconstruction in thoracic CT scans with application to nodule detection. *IEEE Transactions on Medical Imaging* 24.(4) (2005), 486–499 (cited on pages 21, 22).

[5] C. Alonso-Montes, D. L. Vilariño, P. Dudek, and M. G. Penedo. Fast retinal vessel tree extraction: A pixel parallel approach. *International Journal of Circuit Theory and Applications* 36.(5-6) (2008), 641–651 (cited on pages 68, 78).

[6] E. Aptoula and S. Lefevre. A comparative study on multivariate mathematical morphology. *Pattern Recognition* 40.(11) (2007), 2914–2929 (cited on page 82).

[7] E. Aptoula and S. Lefevre. On lexicographical ordering in multivariate mathematical morphology. *Pattern Recognition Letters* 29.(2) (2008), 109–118 (cited on page 82).

[8] G. Argenziano, C. Longo, A. Cameron, S. Cavicchini, J.-Y. Gourhant, A. Lallas, I. McColl, C. Rosendahl, L. Thomas, D. Tiodorovic-Zivkovic, et al. Blue-black rule: a simple dermoscopic clue to recognize pigmented nodular melanoma. *British Journal of Dermatology* 165.(6) (2011), 1251–1255 (cited on page 99).

[9] F. Argüello, D. L. Vilariño, D. B. Heras, and A. Nieto. GPU-based segmentation of retinal blood vessels. *Journal of Real-Time Image Processing* (2014), 1–10 (cited on pages 68, 74, 77–80).

[10] J. Ashmead. Morlet Wavelets in Quantum Mechanics. *Quanta* 1.(1) (2012), 58–70 (cited on page 31).

[11] G. Azzopardi, N. Strisciuglio, M. Vento, and N. Petkov. Trainable COSFIRE filters for vessel delineation with application to retinal images. *Medical image analysis* 19.(1) (2015), 46–57 (cited on pages 21, 22, 26, 44, 46, 47, 68, 69, 74, 78, 79).

[12] M. Baczyński and B. Jayaram. *Fuzzy Implications*. Volume 231. Studies in Fuzziness and Soft Computing. Springer, 2008 (cited on pages 9, 10).

[13] M. Baczyński, B. Jayaram, S. Massanet, and J. Torrens. Fuzzy implications: past, present, and future. *Springer Handbook of Computational Intelligence*. Springer, 2015, p. 183–202 (cited on pages 9, 10).

[14] X. Bai, T. Wang, and F. Zhou. Linear feature detection based on the multi-scale, multi-structuring element, grey-level hit-or-miss transform. *Computers & Electrical Engineering* 46 (2015), 487–499 (cited on pages 50, 51, 55, 59–63, 71, 72).

[15] P. Bankhead, C. N. Scholfield, J. G. McGeown, and T. M. Curtis. Fast retinal vessel detection and measurement using wavelets and edge location refinement. *PloS one* 7.(3) (2012), e32435 (cited on pages 68, 77–80).

[16] C. Barat, C. Ducottet, et al. Pattern matching using morphological probing. *2003 International Conference on Image Processing, ICIP 2003*. Volume 1. IEEE. 2003, p. I–369 (cited on pages 59–64, 71, 72).

[17] V. Barnett. The ordering of multivariate data. *Journal of the Royal Statistical Society. Series A (General)* 139.(3) (1976), 318–355 (cited on page 82).

[18] M. Barva, J. Kybic, J.-M. Mari, C. Cachard, and V. Hlavác. Automatic localization of curvilinear object in 3D ultrasound images. *Medical Imaging*. International Society for Optics and Photonics. 2005, p. 455–462 (cited on pages 21–23, 33, 47).

[19] N. Batool and R. Chellappa. Fast detection of facial wrinkles based on Gabor features using image morphology and geometric constraints. *Pattern Recognition* 48.(3) (2015), 642–658 (cited on pages 21–24, 31, 46).

[20] N. Batool and R. Chellappa. Modeling and detection of wrinkles in aging human faces using marked point processes. *Computer Vision–ECCV 2012. Workshops and Demonstrations*. Springer. 2012, p. 178–188 (cited on pages 21–24, 32, 46).

[21] N. Batool, S. Taheri, and R. Chellappa. Assessment of facial wrinkles as a soft biometrics. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. IEEE. 2013, p. 1–7 (cited on page 23).

[22] C. Bauer and H. Bischof. A novel approach for detection of tubular objects and its application to medical image analysis. *Pattern Recognition*. Springer, 2008, p. 163–172 (cited on pages 21, 36, 47).

[23] C. Bauer, H. Bischof, and R. Beichel. Segmentation of airways based on gradient vector flow. *International workshop on pulmonary image analysis, Medical image computing and computer assisted intervention*. Citeseer. 2009, p. 191–201 (cited on pages 21, 36, 45).

[24] C. Bauer, M. Eberlein, and R. R. Beichel. Graph-Based Airway Tree Reconstruction From Chest CT Scans: Evaluation of Different Features on Five Cohorts. *IEEE transactions on medical imaging* 34.(5) (2015), 1063–1076 (cited on pages 21, 40, 45, 46).

[25] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua. Supervised feature learning for curvilinear structure segmentation. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*. Springer, 2013, p. 526–533 (cited on pages 24, 38).

[26] T. Behrens, K. Rohr, and H. S. Stiehl. Robust segmentation of tubular structures in 3-D medical images by parametric object detection and tracking. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 33.(4) (2003), 554–561 (cited on page 46).

[27] G. Beliakov, A. Pradera, and T. Calvo. *Aggregation functions: A guide for practitioners*. Volume 221. Springer, 2007 (cited on page 9).

[28] X. Benavent, E. Dura, F. Vegara, and J. Domingo. Mathematical Morphology for Color Images: An Image-Dependent Approach. *Mathematical Problems in Engineering* 2012 (2012). Article ID 678326, 1–18 (cited on pages 83, 85, 92–97).

[29] F. Benmansour and L. D. Cohen. Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision* 92.(2) (2011), 192–210 (cited on pages 21, 37).

[30] S. Berlemont and J.-C. Olivo-Marin. Combining local filtering and multiscale analysis for edge, ridge, and curvilinear objects detection. *IEEE Transactions on Image Processing* 19.(1) (2010), 74–84 (cited on page 31).

[31] S. Berlemont, R. Tournebize, A. Bensimon, and J.-C. Olivo-Marin. Detection of full length microtubules in live microscopy images. *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2008*. IEEE. 2008, p. 851–854 (cited on page 24).

[32] P. Bibiloni, M. González-Hidalgo, and S. Massanet. A Survey on Curvilinear Object Segmentation in Multiple Applications. *Pattern Recognition* 60 (2016), 949–970 (cited on page 50).

[33] P. Bibiloni, M. González-Hidalgo, and S. Massanet. Retinal Vessel Detection Based on Fuzzy Morphological Line Enhancement. *Advances in Artificial Intelligence*. Lecture Notes in Computer Science. Springer, 2015, p. 61–70 (cited on pages 21, 29, 44).

[34] P. Bibiloni, M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. Mayor-Torrens t-norms in the Fuzzy Mathematical Morphology and Their Applications. *Fuzzy Logic and Information Fusion*. Springer, 2016, p. 201–235 (cited on page 75).

[35] V. Bismuth, R. Vaillant, H. Talbot, and L. Najman. Curvilinear structure enhancement with the polygonal path image-application to guide-wire segmentation in X-ray fluoroscopy. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*. Springer, 2012, p. 9–16 (cited on pages 21–23).

[36] I. Bloch. Fuzzy sets for image processing and understanding. *Fuzzy Sets and Systems* 281 (2015), 280–291 (cited on page 11).

[37] I. Bloch. Fuzzy sets in image processing. *Proceedings of the 1994 ACM symposium on Applied computing*. ACM. 1994, p. 175–179 (cited on page 2).

[38] I. Bloch and H. Maître. Fuzzy mathematical morphologies: a comparative study. *Pattern recognition* 28.(9) (1995), 1341–1387 (cited on pages 11, 89).

[39] R. Bock, J. Meier, L. G. Nyúl, J. Hornegger, and G. Michelson. Glaucoma risk index: automated glaucoma detection from color fundus images. *Medical image analysis* 14.(3) (2010), 471–481 (cited on page 72).

[40] S. Born, D. Iwamaru, M. Pfeifle, and D. Bartz. Three-step segmentation of the lower airways with advanced leakage-control. *Proc. of Second International Workshop on Pulmonary Image Analysis*. 2009, p. 239–249 (cited on pages 21, 26, 45, 47).

[41] A. Bouchet, P. Alonso, J. I. Pastore, S. Montes, and I. Díaz. Fuzzy mathematical morphology for color images defined by fuzzy preference relations. *Pattern Recognition* 60 (2016), 720–733 (cited on page 83).

[42] A. Budai, R. Bock, A. Maier, J. Hornegger, and G. Michelson. Robust vessel segmentation in fundus images. *International journal of biomedical imaging* 2013 (2013). Article ID 1548600, 1–11 (cited on pages 21, 37).

[43] J. Candamo and D. Goldgof. Wire detection in low-altitude, urban, and low-quality video frames. *19th International Conference on Pattern Recognition, ICPR 2008*. IEEE. 2008, p. 1–4 (cited on pages 24, 27, 45).

[44] J. Candamo, R. Kasturi, D. Goldgof, and S. Sarkar. Detection of thin lines using low-quality video from low-altitude aircraft in urban settings. *IEEE Transactions on Aerospace and Electronic Systems* 45.(3) (2009), 937–949 (cited on pages 20, 24, 27).

[45] M. J. Carlotto. Enhancement of low-contrast curvilinear features in imagery. *IEEE Transactions on Image Processing* 16.(1) (2007), 221–228 (cited on pages 23, 40).

[46] S. Chambon and J.-M. Moliard. Automatic road pavement assessment with image processing: Review and comparison. *International Journal of Geophysics* 2011 (2011). Article ID 989354, 1–20 (cited on pages 18, 19, 23, 49).

[47] J. Chanussot and P. Lambert. Total ordering based on space filling curves for multivalued morphology. *Computational Imaging and Vision* 12 (1998), 51–58 (cited on pages 83, 92–97).

[48] T. Chanwimaluang and G. Fan. An efficient blood vessel detection algorithm for retinal images using local entropy thresholding. *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS'03*. Volume 5. IEEE. 2003, p. V–21 (cited on pages 68, 78, 79).

[49] D. Chaudhuri and A. Samal. An automatic bridge detection technique for multispectral images. *IEEE Transactions on Geoscience and Remote Sensing* 46.(9) (2008), 2720–2727 (cited on pages 23, 40).

[50] S. Chaudhuri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum. Detection of blood vessels in retinal images using two-dimensional matched filters. *IEEE Transactions on medical imaging* 8.(3) (1989), 263–269 (cited on pages 21, 22, 27).

[51] O. Chum and J. Matas. Randomized RANSAC with $T_{d,d}$ test. *Proc. British Machine Vision Conference*. Volume 2. 2002, p. 448–457 (cited on page 33).

[52] L. Cohen. The wavelet transform and time-frequency analysis. *Wavelets and signal processing*. Springer, 2003, p. 3–22 (cited on page 31).

[53] F. Cokelaer, H. Talbot, and J. Chanussot. Efficient robust d-dimensional path operators. *IEEE journal of Selected Topics in Signal Processing* 6.(7) (2012), 830–839 (cited on pages 29, 50).

[54] G. O. Cula, P. R. Bargo, A. Nkengne, and N. Kollias. Assessing facial wrinkles: automatic detection and quantification. *Skin Research and Technology* 19.(1) (2013), e243–e251 (cited on pages 21–24, 31, 61).

[55] B. De Baets. A fuzzy morphology: a logical approach. *Uncertainty Analysis in Engineering and Sciences: Fuzzy Logic, Statistics, and Neural Network Approach*. Springer, 1998, p. 53–67 (cited on pages 2, 11, 12, 29, 92).

[56] V. De Witte, S. Schulte, M. Nachtegael, T. Mélange, and E. E. Kerre. A lattice-based approach to mathematical morphology for greyscale and colour images. *Computational Intelligence Based on Lattice Theory*. Springer, 2007, p. 129–148 (cited on page 83).

[57]   C. R. Dillabaugh, K. O. Niemann, and D. E. Richardson. Semi-automated extraction of rivers from digital imagery. *GeoInformatica* 6.(3) (2002), 263–284 (cited on page 23).

[58]   B. Al-Diri, A. Hunter, and D. Steel. An active contour model for segmenting and measuring retinal vessels. *IEEE Transactions on Medical Imaging* 28.(9) (2009), 1488–1497 (cited on pages 21, 34, 44, 47).

[59]   D. L. Donoho and X. Huo. Beamlets and Multiscale Image Analysis. *Multiscale and Multiresolution Methods: Theory and Applications* 20 (2002), 149–196 (cited on pages 31, 47).

[60]   K. J. Draper, C. C. Blake, L. Gowman, D. B. Downey, and A. Fenster. An algorithm for automatic needle localization in ultrasound-guided breast biopsies. *Medical Physics* 27.(8) (2000), 1971–1979 (cited on pages 22, 32).

[61]   A. Dufour, O. Tankyevych, B. Naegel, H. Talbot, C. Ronse, J. Baruthio, P. Dokládal, and N. Passat. Filtering and segmentation of 3D angiographic data: Advances based on mathematical morphology. *Medical image analysis* 17.(2) (2013), 147–164 (cited on page 21).

[62]   R. Estrada, C. Tomasi, M. T. Cabrera, D. K. Wallace, S. F. Freedman, and S. Farsiu. Exploratory Dijkstra forest based automatic vessel segmentation: applications in video indirect ophthalmoscopy (VIO). *Biomedical optics express* 3.(2) (2012), 327–339 (cited on page 69).

[63]   A. Ferraz, C. Mallet, and N. Chehata. Large scale road network extraction in forested moutainous areas using airborne laser scanning data. *2014 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2014, p. 4315–4318 (cited on pages 21–23, 32, 36, 46).

[64]   C. Fetita, M. Ortner, P.-Y. Brillet, F. Prêteux, P. Grenier, et al. A morphological-aggregative approach for 3D segmentation of pulmonary airways from generic MSCT acquisitions. *Proc. of Second International Workshop on Pulmonary Image Analysis*. 2009, p. 215–226 (cited on pages 21, 28, 45).

[65]   M. Feuerstein, T. Kitasaka, and K. Mori. Adaptive branch tracing and image sharpening for airway tree extraction in 3-D chest CT. *Proc. Second International Workshop on Pulmonary Image Analysis*. 2009, p. 273–284 (cited on pages 21, 39, 45).

[66]   O. Fialka and M. Čadík. FFT and convolution performance in image filtering on GPU. *Tenth International Conference on Information Visualization, IV 2006*. IEEE. 2006, p. 609–614 (cited on page 26).

[67]   A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *Medical Image Computing and Computer-Assisted Intervention—MICCAI'98*. Springer, 1998, p. 130–137 (cited on pages 21, 22, 26, 36, 37).

[68]   M. M. Fraz, S. Barman, P. Remagnino, A. Hoppe, A. Basit, B. Uyyanonvara, A. R. Rudnicka, and C. G. Owen. An approach to localize the retinal blood vessels using bit planes and centerline detection. *Computer methods and programs in biomedicine* 108.(2) (2012), 600–616 (cited on pages 21, 28, 44).

[69]   M. M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. R. Rudnicka, C. G. Owen, and S. A. Barman. Blood vessel segmentation methodologies in retinal images–a survey. *Computer methods and programs in biomedicine* 108.(1) (2012), 407–433 (cited on pages 2, 18, 21, 22, 49, 69).

[70] M. van Ginkel, C. L. Hendriks, and L. J. van Vliet. *A short introduction to the Radon and Hough transforms and how they relate to each other*. Technical report QI-2004-01. 2004, pages 1–9 (cited on page 30).

[71] M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. A fuzzy morphological hit-or-miss transform for grey-level images: A new approach. *Fuzzy Sets and Systems* 286 (2016), 30–65 (cited on page 13).

[72] M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. A fuzzy filter for high-density salt and pepper noise removal. *Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2013*. Volume 8109. Lecture Notes in Computer Science. Springer. 2013, p. 70–79 (cited on page 102).

[73] M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. Improving salt and pepper noise removal using a fuzzy mathematical morphology-based filter. *Applied Soft Computing* 63 (2018), 167–180 (cited on page 102).

[74] M. González-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera. On the choice of the pair conjunction–implication into the fuzzy morphological edge detector. *IEEE Transactions on Fuzzy Systems* 23.(4) (2015), 872–884 (cited on page 73).

[75] R. C. Gonzalez, R. E. Woods, and S. L. Eddins. *Digital Image Processing Using MATLAB*. 2nd. Gatesmark Publishing, 2004 (cited on pages 73, 92, 107).

[76] J. Goutsias, H. J. Heijmans, and K. Sivakumar. Morphological operators for image sequences. *Computer Vision and Image Understanding* 62.(3) (1995), 326–346 (cited on pages 90, 91).

[77] J. J. van de Gronde and J. B. Roerdink. Group-invariant colour morphology based on frames. *IEEE Transactions on Image Processing* 23.(3) (2014), 1276–1288 (cited on pages 82, 88).

[78] C. Gu. Multivalued morphology and its application in moving object segmentation and tracking. *Mathematical morphology and its applications to image and signal processing*. Springer, 1996, p. 345–352 (cited on pages 82, 92–97).

[79] A. Haas, G. Matheron, and J. Serra. Morphologie mathématique et granulométries en place. *Annales des mines*. (11). 1967, p. 736–753 (cited on page 11).

[80] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence* 9.(4) (1987), 532–550 (cited on page 81).

[81] H. Heijmans, M. Buckley, and H. Talbot. Path openings and closings. *Journal of Mathematical Imaging and Vision* 22.(2-3) (2005), 107–119 (cited on pages 29, 50, 59).

[82] N. Honnorat, R. Vaillant, and N. Paragios. Guide-wire extraction through perceptual organization of local segments in fluoroscopic images. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*. Springer, 2010, p. 440–448 (cited on pages 22, 23, 38, 47).

[83] A. Hoover, V. Kouznetsova, and M. Goldbaum. Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging* 19.(3) (2000), 203–210 (cited on pages 21, 27, 43, 44, 67, 68, 76, 79).

[84] A. Huang, G. M. Nielson, A. Razdan, G. E. Farin, D. P. Baluch, and D. G. Capco. Thin structure segmentation and visualization in three-dimensional biomedical images: a shape-based approach. *IEEE Transactions on Visualization and Computer Graphics* 12.(1) (2006), 93–102 (cited on pages 26, 37, 47).

[85] D.-S. Huang, W. Jia, and D. Zhang. Palmprint verification based on principal lines. *Pattern Recognition* 41.(4) (2008), 1316–1328 (cited on pages 24, 25, 30).

[86] *Informe de conclusiones. MELANOMA VISIÓN 360°: Diálogos entre pacientes y profesionales.* Madrid, 2015 (cited on page 99).

[87] H. J. Jelinek, M. J. Cree, D. Worsley, A. Luckie, and P. Nixon. An automated microaneurysm detector as a tool for identification of diabetic retinopathy in rural optometric practice. *Clinical and Experimental Optometry* 89.(5) (2006), 299–305 (cited on page 72).

[88] A. F. Jerant, J. T. Johnson, C. Sheridan, T. J. Caffrey, et al. Early detection and treatment of skin cancer. *American family physician* 62.(2) (2000), 357–386 (cited on page 99).

[89] A. Jiaqiu, Q. Xiangyang, Y. Weidong, D. Yunkai, L. Fan, S. Li, and J. Yafei. A novel ship wake CFAR detection algorithm based on SCR enhancement and normalized hough transform. *Geoscience and Remote Sensing Letters, IEEE* 8.(4) (2011), 681–685 (cited on page 23).

[90] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision* 1.(4) (1988), 321–331 (cited on page 34).

[91] K. Kiani and A. R. Sharafat. E-shaver: An improved DullRazor® for digitally removing dark and light-colored hairs in dermoscopic images. *Computers in biology and medicine* 41.(3) (2011), 139–145 (cited on page 100).

[92] C. Kirbas and F. Quek. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys (CSUR)* 36.(2) (2004), 81–121 (cited on pages 2, 18, 21).

[93] T. Kitasaka, K. Mori, Y. Suenaga, J.-i. Hasegawa, and J.-i. Toriwaki. A method for segmenting bronchial trees from 3D chest X-ray CT images. *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2003*. Springer, 2003, p. 603–610 (cited on page 21).

[94] S. Klemenjak, B. Waske, S. Valero, and J. Chanussot. Automatic detection of rivers in high-resolution SAR data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5.(5) (2012), 1364–1372 (cited on pages 23, 50).

[95] T. Kohler, A. Budai, M. F. Kraus, J. Odstrcilik, G. Michelson, and J. Hornegger. Automatic no-reference quality assessment for retinal fundus images using vessel segmentation. *IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS), 2013*. IEEE. 2013, p. 95–100 (cited on page 21).

[96] T. M. Koller, G. Gerig, G. Szekely, and D. Dettwiler. Multiscale detection of curvilinear structures in 2-D and 3-D image data. *Fifth International Conference on Computer Vision, 1995. Proceedings*. IEEE. 1995, p. 864–869 (cited on page 26).

[97] A. Kong, D. Zhang, and M. Kamel. A survey of palmprint recognition. *Pattern Recognition* 42.(7) (2009), 1408–1418 (cited on pages 18, 24, 49).

[98] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model-based detection of tubular structures in 3D images. *Computer vision and image understanding* 80.(2) (2000), 130–171 (cited on pages 21, 22, 69).

[99]   V. Krylov, J. D. Nelson, et al. Stochastic extraction of elongated curvilinear structures with applications. *IEEE Transactions on Image Processing* 23.(12) (2014), 5360–5373 (cited on pages 30, 50, 59–63, 71, 72).

[100] P. M. Kulkarni, E. Barton, M. Savelonas, R. Padmanabhan, Y. Lu, K. Trett, W. Shain, J. L. Leasure, and B. Roysam. Quantitative 3-D analysis of GFAP labeled astrocytes from fluorescence confocal images. *Journal of neuroscience methods* 246 (2015), 38–51 (cited on pages 24, 37).

[101] C. Lacoste, X. Descombes, and J. Zerubia. Point processes for unsupervised line network extraction in remote sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.(10) (2005), 1568–1579 (cited on pages 23, 32).

[102] C. Lacoste, G. Finet, and I. E. Magnin. Coronary tree extraction from X-ray angiograms using marked point processes. *3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006*. IEEE. 2006, p. 157–160 (cited on pages 21, 32).

[103] I. Laptev, H. Mayer, T. Lindeberg, W. Eckstein, C. Steger, and A. Baumgartner. Automatic extraction of roads from aerial images based on scale space and snakes. *Machine Vision and Applications* 12.(1) (2000), 23–31 (cited on page 19).

[104] M. W. Law and A. C. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. *Computer Vision–ECCV 2008*. Springer, 2008, p. 368–382 (cited on pages 21, 37).

[105] J. Lee and A. P. Reeves. Segmentation of the airway tree from chest CT using local volume of interest. *Proc. of Second International Workshop on Pulmonary Image Analysis*. 2009, p. 273–284 (cited on pages 21, 39, 45, 46).

[106] T. Lee, V. Ng, R. Gallagher, A. Coldman, and D. McLean. Dullrazor®: A software approach to hair removal from images. *Computers in Biology and Medicine* 27.(6) (1997), 533–543 (cited on pages 100, 104, 105).

[107] C. Lemaitre, M. Perdoch, A. Rahmoune, J. Matas, and J. Mitéran. Detection and matching of curvilinear structures. *Pattern recognition* 44.(7) (2011), 1514–1527 (cited on pages 24, 27).

[108] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea. A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical image analysis* 13.(6) (2009), 819–845 (cited on pages 18, 19, 49).

[109] O. Lézoray. Complete lattice learning for multivariate mathematical morphology. *Journal of Visual Communication and Image Representation* 35 (2016), 220–235 (cited on page 83).

[110] H. Li, W. Hsu, M. L. Lee, and H. Wang. A piecewise Gaussian model for profiling and differentiating retinal vessels. *2003 International Conference on Image Processing, ICIP 2003*. Volume 1. IEEE. 2003, p. I–1069 (cited on pages 21, 33, 47, 64).

[111] F. Liu, G. Xu, Y. Yang, X. Niu, and Y. Pan. Novel approach to pavement cracking automatic detection based on segment extending. *International Symposium on Knowledge Acquisition and Modeling, 2008. KAM'08*. IEEE. 2008, p. 610–614 (cited on pages 21–23, 40).

[112] P. C. P. Lo, J. Sporring, and M. de Bruijne. Multiscale vessel-guided airway tree segmentation. *The Second International Workshop on Pulmonary Image Analysis*. 2009, p. 323–332 (cited on pages 21, 22, 38, 45–47).

[113] P. Lo, B. Van Ginneken, J. M. Reinhardt, T. Yavarna, P. de Jong, B. Irving, C. Fetita, M. Ortner, R. Pinho, J. Sijbers, et al. Extraction of airways from CT (EXACT'09). *IEEE Transactions on Medical Imaging* 31.(11) (2012), 2093–2107 (cited on pages 2, 18, 20–22, 42–45, 49).

[114] C. P. Loizou, C. S. Pattichis, M. Pantziaris, T. Tyllis, and A. Nicolaides. Snakes based segmentation of the common carotid artery intima media. *Medical & Biological Engineering & Computing* 45.(1) (2007), 35–49 (cited on page 61).

[115] H. Lorentzen, K. Weismann, L. Secher, C. Sand Petersen, and F. Grønhøj Larsen. The dermatoscopic ABCD rule does not improve diagnostic accuracy of malignant melanoma. *Acta Dermatovenereológica* 79 (1999), 469–472 (cited on page 99).

[116] G. Louverdis, I. Andreadis, and P. Tsalides. New fuzzy model for morphological colour image processing. *IEE Proceedings-Vision, Image and Signal Processing* 149.(3) (2002), 129–139 (cited on pages 83, 92–97).

[117] D. Maji, A. Santara, P. Mitra, and D. Sheet. Ensemble of deep convolutional neural networks for learning to detect retinal vessels in fundus images. *arXiv preprint arXiv:1603.04833* (2016) (cited on page 68).

[118] C. Mariño, M. G. Penedo, M. Penas, M. J. Carreira, and F. Gonzalez. Personal authentication using digital retinal images. *Pattern Analysis and Applications* 9.(1) (2006), 21–33 (cited on page 21).

[119] H. Mayer, S. Hinz, U. Bacher, and E. Baltsavias. A test of automatic road extraction approaches. *International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences* 36.(3) (2006), 209–214 (cited on pages 23, 44).

[120] J. McCarthy and P. J. Hayes. Some Philosophical Problems from the Standpoint of Artificial Intelligence. *Machine Intelligence 4*. Edited by B. Meltzer and D. Michie. reprinted in McC90. Edinburgh University Press, 1969, p. 463–502 (cited on page 1).

[121] T. McInerney and D. Terzopoulos. T-snakes: Topology adaptive snakes. *Medical image analysis* 4.(2) (2000), 73–91 (cited on page 34).

[122] R. Medina-Carnicer, R. Munoz-Salinas, E. Yeguas-Bolivar, and L. Diaz-Mas. A novel method to look for the hysteresis thresholds for the Canny edge detector. *Pattern Recognition* 44.(6) (2011), 1201–1211 (cited on page 13).

[123] E. Meijering, M. Jacob, J.-C. Sarria, P. Steiner, H. Hirling, and M. Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A* 58.(2) (2004), 167–176 (cited on pages 24, 37).

[124] J. B. Mena. State of the art on automatic road extraction for GIS update: a novel classification. *Pattern Recognition Letters* 24.(16) (2003), 3037–3058 (cited on pages 18, 23, 49).

[125] A. M. Mendonca and A. Campilho. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Transactions on Medical Imaging* 25.(9) (2006), 1200–1213 (cited on pages 21, 22, 28, 44, 68, 69).

[126] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira. PH 2 - A dermoscopic image database for research and benchmarking. *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2013, p. 5437–5440 (cited on pages 102, 104).

[127] 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), 246–261 (cited on pages 50, 59–64, 70–72).

[128] 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.(2) (2018), 304–317 (cited on pages 50, 59).

[129] M. Nachtegael and E. E. Kerre. Classical and fuzzy approaches towards mathematical morphology. *Fuzzy techniques in image processing*. Springer, 2000, p. 3–57 (cited on pages 11, 12, 28, 91).

[130] Y. Nakano, S. Muro, H. Sakai, T. Hirai, K. Chin, M. Tsukino, K. Nishimura, H. Itoh, P. D. Pare, J. C. Hogg, et al. Computed tomographic measurements of airway dimensions and emphysema in smokers: correlation with lung function. *American journal of respiratory and critical care medicine* 162.(3) (2000), 1102–1108 (cited on page 22).

[131] H. Narasimha-Iyer, V. Mahadevan, J. M. Beach, and B. Roysam. Improved detection of the central reflex in retinal vessels using a generalized dual-Gaussian model and robust hypothesis testing. *IEEE Transactions on Information Technology in Biomedicine* 12.(3) (2008), 406–410 (cited on pages 21, 32).

[132] N. H. Nguyen, T. K. Lee, and M. S. Atkins. Segmentation of light and dark hair in dermoscopic images: a hybrid approach using a universal kernel. *Medical Imaging 2010: Image Processing*. Volume 7623. SPIE. International Society for Optics and Photonics. 2010, p. 1–8 (cited on page 100).

[133] M. Niemeijer, J. Staal, B. van Ginneken, M. Loog, and M. D. Abramoff. Comparative study of retinal vessel segmentation methods on a new publicly available database. *Medical Imaging 2004: Image Processing*. Volume 5370. International Society for Optics and Photonics. 2004, p. 648–657 (cited on page 68).

[134] J. Odstrcilik, R. Kolar, V. Harabis, and R. Tornow. Classification-based blood vessel segmentation in retinal images. *Computational Vision and Medical Image Processing V*. CRC Press. Taylor & Francis, 2015, p. 95 (cited on page 69).

[135] J. Odstrcilik, R. Kolar, A. Budai, J. Hornegger, J. Jan, J. Gazarek, T. Kubena, P. Cernosek, O. Svoboda, and E. Angelopoulou. Retinal vessel segmentation by improved matched filtering: evaluation on a new high-resolution fundus image database. *IET Image Processing* 7.(4) (2013), 373–383 (cited on page 69).

[136] H. Oliveira and P. L. Correia. Automatic road crack detection and characterization. *IEEE Transactions on Intelligent Transportation Systems* 14.(1) (2013), 155–168 (cited on pages 22, 23, 39).

[137] H. Oliveira and P. L. Correia. Identifying and retrieving distress images from road pavement surveys. *15th IEEE International Conference on Image Processing, ICIP 2008*. IEEE. 2008, p. 57–60 (cited on pages 21–23).

[138] D. Padfield, J. Rittscher, and B. Roysam. Spatio-temporal cell segmentation and tracking for automated screening. *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2008*. IEEE. 2008, p. 376–379 (cited on page 41).

[139] N. Padoy and G. D. Hager. Deformable Tracking of Textured Curvilinear Objects. *Proceedings of the British Machine Vision Conference*. British Machine Vision Association and Society for Pattern Recognition. 2012, p. 1–11 (cited on pages 23, 34).

[140] D. Palti-Wasserman, A. M. Brukstein, and R. P. Beyar. Identifying and tracking a guide wire in the coronary arteries during angioplasty from X-ray images. *IEEE Transactions on Biomedical Engineering* 44.(2) (1997), 152–164 (cited on pages 21–23, 30, 46).

[141] C. Panagiotakis, E. Kokinou, and A. Sarris. Curvilinear structure enhancement and detection in geophysical images. *IEEE Transactions on Geoscience and Remote Sensing* 49.(6) (2011), 2040–2048 (cited on pages 24, 27).

[142] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing* 39.(3) (1987), 355–368 (cited on pages 14, 64).

[143] A. Pradera and E. Trillas. Aggregation operators from the ancient NC and EM point of view. *Kybernetika* 42.(3) (2006), 243–260 (cited on page 10).

[144] J. Pu, C. Fuhrman, W. F. Good, F. C. Sciurba, and D. Gur. A differential geometric approach to automated segmentation of human airway tree. *IEEE Transactions on Medical Imaging* 30.(2) (2011), 266–278 (cited on pages 21, 22, 35, 46).

[145] X. Qian, M. P. Brennan, D. P. Dione, W. L. Dobrucki, M. P. Jackowski, C. K. Breuer, A. J. Sinusas, and X. Papademetris. A non-parametric vessel detection method for complex vascular structures. *Medical Image Analysis* 13.(1) (2009), 49–61 (cited on pages 21, 40).

[146] E. Ricci and R. Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Transactions on Medical Imaging* 26.(10) (2007), 1357–1365 (cited on pages 21, 38).

[147] Y. Rouchdy and L. D. Cohen. Image segmentation by geodesic voting. Application to the extraction of tree structures from confocal microscope images. *19th International Conference on Pattern Recognition, ICPR 2008*. IEEE. 2008, p. 1–5 (cited on pages 24, 40).

[148] S. Roychowdhury, D. D. Koozekanani, and K. K. Parhi. Blood Vessel Segmentation of Fundus Images by Major Vessel Extraction and Subimage Classification. *IEEE Journal of Biomedical and Health Informatics* 19.(3) (2015), 1118–1128 (cited on pages 78, 79).

[149] S. Roychowdhury, D. Koozekanani, and K. Parhi. Iterative Vessel Segmentation of Fundus Images. *IEEE Transactions on Biomedical Engineering* 62.(7) (2015), 1738–1749 (cited on pages 21, 29, 44, 46, 68, 77–79).

[150] T. Saar and O. Talvik. Automatic asphalt pavement crack detection and classification using neural networks. *12th Biennial Baltic Electronics Conference, BEC 2010*. IEEE. 2010, p. 345–348 (cited on pages 21–23, 38).

[151] S. A. Salem, N. M. Salem, and A. K. Nandi. Segmentation of retinal blood vessels using a novel clustering algorithm (RACAL) with a partial supervision strategy. *Medical & biological engineering & computing* 45.(3) (2007), 261–273 (cited on pages 21, 38, 44, 47).

[152] J. N. Sanders-Reed, D. J. Yelton, C. C. Witt, and R. R. Galetti. Passive obstacle detection system (PODS) for wire detection. *Enhanced and Synthetic Vision 2009*. Volume 7328. SPIE. International Society for Optics and Photonics. 2009, p. 1–12 (cited on pages 24, 35, 45, 63).

[153] L. J. Sartor and A. R. Weeks. Morphological operations on color images. *Journal of Electronic Imaging* 10.(2) (2001), 548–559 (cited on pages 83, 92–97).

[154] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis. Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical image analysis* 2.(2) (1998), 143–168 (cited on pages 21, 47).

[155] Y. Sato, C.-F. Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue classification based on 3D local intensity structures for volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 6.(2) (2000), 160–180 (cited on pages 26, 37).

[156] C.-B. Schönlieb. *Partial Differential Equation Methods for Image Inpainting*. Volume 29. Cambridge University Press, 2015 (cited on page 102).

[157] J. Serra. *Image analysis and mathematical morphology. Vol. 1*. Image Analysis and Mathematical Morphology. Academic press, 1982 (cited on pages 11, 92).

[158] J. Serra. *Image analysis and mathematical morphology. Vol. 2*. Image Analysis and Mathematical Morphology. Academic press, 1984 (cited on pages 11, 82).

[159] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences* 93.(4) (1996), 1591–1595 (cited on page 37).

[160] E. M. Sigurðsson, S. Valero, J. A. Benediktsson, J. Chanussot, H. Talbot, and E. Stefánsson. Automatic retinal vessel extraction based on directional mathematical morphology and fuzzy classification. *Pattern Recognition Letters* 47 (2014), 164–171 (cited on pages 21, 29, 50, 71).

[161] J. V. Soares, J. J. Leandro, R. M. Cesar Jr, H. F. Jelinek, and M. J. Cree. Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging* 25.(9) (2006), 1214–1222 (cited on pages 21, 26, 38, 44, 47, 68, 73, 74, 78, 79).

[162] J. Staal, M. D. Abràmoff, M. Niemeijer, M. Viergever, B. Van Ginneken, et al. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging* 23.(4) (2004), 501–509 (cited on pages 20–22, 37, 43, 44, 58, 68, 70, 72, 73, 76, 78).

[163] C. Steger. An unbiased detector of curvilinear structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.(2) (1998), 113–125 (cited on pages 23, 36).

[164] S. R. Sternberg. Grayscale morphology. *Computer vision, graphics, and image processing* 35.(3) (1986), 333–355 (cited on pages 58, 59).

[165] R. Stoica, X. Descombes, and J. Zerubia. A Gibbs point process for road extraction from remotely sensed images. *International Journal of Computer Vision* 57.(2) (2004), 121–136 (cited on pages 21–23, 32, 36, 46, 47).

[166] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba. Automation of pavement surface crack detection using the continuous wavelet transform. *IEEE International Conference on Image Processing*. IEEE. 2006, p. 3037–3040 (cited on pages 22, 23, 31).

[167] P. Sussner and M. E. Valle. Classification of fuzzy mathematical morphologies based on concepts of inclusion measure and duality. *Journal of Mathematical Imaging and Vision* 32.(2) (2008), 139–159 (cited on pages 10–12).

[168] H. Talbot and B. Appleton. Efficient complete and incomplete path openings and closings. *Image and Vision Computing* 25.(4) (2007), 416–425 (cited on pages 29, 50, 59–64, 71, 72).

[169] S. Tanovic. Line Detection in Low and High Altitude Rotorcraft Flights Infrared Video. Master's thesis. University of Granada, 2011 (cited on pages 24, 45).

[170] M. Tkalcic and J. F. Tasic. Colour spaces: perceptual, historical and applicational background. *The IEEE Region 8 EUROCON 2003: Computer as a Tool*. Volume 1. IEEE, 2003, p. 304–308 (cited on page 9).

[171] M. T. B. Toossi, H. R. Pourreza, H. Zare, M.-H. Sigari, P. Layegh, and A. Azimi. An effective hair removal algorithm for dermoscopy images. *Skin Research and Technology* 19.(3) (2013), 230–235 (cited on pages 100, 104, 105).

[172] J. Tschirren, T. Yavarna, and J. M. Reinhardt. Airway segmentation framework for clinical environments. *Proc. of Second International Workshop on Pulmonary Image Analysis*. 2009, p. 227–238 (cited on pages 21, 22, 26, 39, 45–47).

[173] F. Tupin, B. Houshmand, and M. Datcu. Road detection in dense urban areas using SAR imagery and the usefulness of multiple views. *IEEE Transactions on Geoscience and Remote Sensing* 40.(11) (2002), 2405–2414 (cited on pages 21–23, 40).

[174] F. Tupin, H. Maitre, J.-F. Mangin, J.-M. Nicolas, and E. Pechersky. Detection of linear features in SAR images: application to road network extraction. *IEEE Transactions on Geoscience and Remote Sensing* 36.(2) (1998), 434–453 (cited on pages 21, 23, 40).

[175] P. Tuveri, L. Ghiani, M. Abukmeil, and G. L. Marcialis. On Combining Edge Detection Methods for Improving BSIF Based Facial Recognition Performances. *International Conference on Articulated Motion and Deformable Objects*. Springer. 2016, p. 108–116 (cited on page 61).

[176] J. A. Tyrrell, E. di Tomaso, D. Fuja, R. Tong, K. Kozak, R. K. Jain, and B. Roysam. Robust 3-D modeling of vasculature imagery using superellipsoids. *IEEE Transactions on Medical Imaging* 26.(2) (2007), 223–237 (cited on pages 21, 22, 33, 41).

[177] S. Valero, J. Chanussot, J. A. Benediktsson, H. Talbot, and B. Waske. Advanced directional mathematical morphology for the detection of the road network in very high resolution remote sensing images. *Pattern Recognition Letters* 31.(10) (2010), 1120–1127 (cited on pages 21–23, 29, 46, 50).

[178] M. E. Valle and R. A. Valente. Mathematical Morphology on the Spherical CIEab Quantale with an Application in Color Image Boundary Detection. *Journal of Mathematical Imaging and Vision* 57.(2) (2017), 183–201 (cited on pages 83, 92–97).

[179] E. M. Van Rikxoort, W. Baggerman, and B. van Ginneken. Automatic segmentation of the airway tree from thoracic CT scans using a multi-threshold approach. *Proc. of Second International Workshop on Pulmonary Image Analysis*. 2009, p. 341–349 (cited on pages 21, 26, 39, 45, 46).

[180] A. Vasilevskiy and K. Siddiqi. Flux maximizing geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.(12) (2002), 1565–1578 (cited on pages 21, 35).

[181] S. Velasco-Forero and J. Angulo. Vector ordering and multispectral morphological image processing. *Advances in Low-Level Color Image Processing*. Springer, 2014, p. 223–239 (cited on pages 81, 83).

[182] S. Velasco-Forero and J. Angulo. Random projection depth for multivariate mathematical morphology. *IEEE Journal of Selected Topics in Signal Processing* 6.(7) (2012), 753–763 (cited on pages 83, 92–97).

[183] M. Vlachos and E. Dermatas. Multi-scale retinal vessel segmentation using line tracking. *Computerized Medical Imaging and Graphics* 34.(3) (2010), 213–227 (cited on pages 21, 40, 44).

[184] G. Wang, S. Huang, and L. Jiao. An automatic bridge detection technique for high resolution SAR images. *2nd Asian-Pacific Conference on Synthetic Aperture Radar, APSAR 2009*. IEEE. 2009, p. 498–501 (cited on page 23).

[185] J. Wang, X. Zhou, J. Lu, J. Lichtman, S.-F. Chang, and S. T. Wong. Dynamic local tracing for 3D axon curvilinear structure detection from microscopic image stack. *4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2007*. IEEE. 2007, p. 81–84 (cited on pages 24, 39).

[186] L. Wang, A. Bhalerao, and R. Wilson. Analysis of retinal vasculature using a multiresolution Hermite model. *IEEE Transactions on Medical Imaging* 26.(2) (2007), 137–152 (cited on pages 21, 33, 44, 46, 47).

[187] X. Wang, T. Heimann, P. Lo, M. Sumkauskaite, M. Puderbach, M. de Bruijne, H. Meinzer, and I. Wegner. Statistical tracking of tree-like tubular structures with efficient branching detection in 3D medical image data. *Physics in medicine and biology* 57.(16) (2012), 5325 (cited on pages 21, 45).

[188] C. Wiedemann, C. Heipke, H. Mayer, and O. Jamet. Empirical evaluation of automatically extracted road axes. *Empirical Evaluation Techniques in Computer Vision* (1998), 172–187 (cited on pages 23, 42, 43).

[189] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin* 1.(6) (1945), 80–83 (cited on pages 41, 77).

[190] G. Wyszecki and W. S. Stiles. *Color science*. Second. Wiley, 2000 (cited on pages 8, 9).

[191] C. Xiao, M. Staring, Y. Wang, D. P. Shamonin, and B. C. Stoel. Multiscale bi-Gaussian filter for adjacent curvilinear structures detection with application to vasculature images. *IEEE Transactions on Image Processing* 22.(1) (2013), 174–188 (cited on pages 33, 47, 64).

[192] F.-Y. Xie, S.-Y. Qin, Z.-G. Jiang, and R.-S. Meng. PDE-based unsupervised repair of hair-occluded information in dermoscopy images of melanoma. *Computerized Medical Imaging and Graphics* 33.(4) (2009), 275–282 (cited on page 100).

[193] G. Xiong, X. Zhou, A. Degterev, L. Ji, and S. T. Wong. Automated neurite labeling and analysis in fluorescence microscopy images. *Cytometry Part A* 69.(6) (2006), 494–505 (cited on pages 24, 37).

[194] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing* 7.(3) (1998), 359–369 (cited on pages 36, 47).

[195] Y. Xu, M. Savelonas, P. Qiu, K. Trett, W. Shain, and B. Roysam. Unsupervised inference of arbor morphology progression for microglia from confocal microscope images. *2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2013, p. 1356–1359 (cited on pages 24, 39).

[196] Z. Xu, U. Bagci, B. Foster, and D. J. Mollura. A hybrid multi-scale approach to automatic airway tree segmentation from CT scans. *2013 IEEE 10th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2013, p. 1308–1311 (cited on pages 21, 37, 45).

[197] N. Yager and A. Amin. Fingerprint classification: a review. *Pattern Analysis and Applications* 7.(1) (2004), 77–93 (cited on pages 18, 49, 60).

[198] L. A. Zadeh. Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems. Edited by G. J. Klir and B. Yuan. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1996. Chapter Fuzzy Sets, p. 19–34 (cited on page 2).

[199] F. Zana and J.-C. Klein. Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Transactions on Image Processing* 10.(7) (2001), 1010–1019 (cited on pages 21, 22, 28, 29, 44, 68, 69, 78).

[200] Y. Zhang, K. Chen, M. Baron, M. A. Teylan, Y. Kim, Z. Song, P. Greengard, and S. T. Wong. A neurocomputational method for fully automated 3D dendritic spine detection and segmentation of medium-sized spiny neurons. *Neuroimage* 50.(4) (2010), 1472–1484 (cited on pages 24, 37).

[201] Y. Zhang, X. Zhou, R. M. Witt, B. L. Sabatini, D. Adjeroh, and S. T. Wong. Dendritic spine detection using curvilinear structure detector and LDA classifier. *Neuroimage* 36.(2) (2007), 346–360 (cited on page 24).

[202] H. Zhao, J. Kumagai, M. Nakagawa, and R. Shibasaki. Semi-automatic road extraction from high-resolution satellite image. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences* 34.(3/A) (2002), 406–411 (cited on pages 21–23, 25, 27).

[203] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang. CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letters* 33.(3) (2012), 227–238 (cited on pages 20–23, 35, 48).

[204] K. Zuiderveld. Contrast limited adaptive histogram equalization. *Graphics gems IV*. Academic Press Professional, Inc. 1994, p. 474–485 (cited on pages 14, 74).

[205] R. Zwiggelaar, T. C. Parr, J. E. Schumm, I. W. Hutt, C. J. Taylor, S. M. Astley, and C. R. Boggis. Model-based detection of spiculated lesions in mammograms. *Medical Image Analysis* 3.(1) (1999), 39–62 (cited on pages 24, 40).