

University of Groningen

Generalized Connected Morphological Operators for Robust Shape Extraction

Ouzounis, Georgios Konstantinou

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2009

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Ouzounis, G. K. (2009). *Generalized Connected Morphological Operators for Robust Shape Extraction*. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

**Generalized Connected Morphological Operators
for
Robust Shape Extraction**

Georgios K. Ouzounis



Netherlands Organisation for Scientific Research

This research project was funded by the Netherlands Organization for Scientific Research under project number 612.065.202.

ISBN: 978-90-367-3698-5

RIJKSUNIVERSITEIT GRONINGEN

**Generalized Connected Morphological Operators
for
Robust Shape Extraction**

Proefschrift

ter verkrijging van het doctoraat in de
Wiskunde en Natuurwetenschappen
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. F. Zwarts,
in het openbaar te verdedigen op
vrijdag 16 januari 2009
om 14:45 uur

door

Georgios Konstantinou Ouzounis

geboren op 13 januari 1977
te Esslingen am Neckar, Duitsland

Promotor : Prof. dr. N. Petkov
Copromotor : Dr. M.H.F. Wilkinson
Beoordelingscommissie : Prof. dr. S. Acton
Prof. dr. P. Maragos
Prof. dr. W. Niessen

ISBN: 978-90-367-3698-5

To the memory of a great man,
Georgios Christou Ouzounis

Contents

1	Introduction	1
1.1	Summary of the Work and Contributions	2
1.1.1	Mask-Based Second-Generation Connectivity	3
1.1.2	Partition-Induced Connectivity	6
1.1.3	Hyperconnectivity	7
1.2	Thesis Organization	8
2	Mask-Based Second-Generation Connectivity and Attribute Filters	11
2.1	Introduction	11
2.2	Theoretical Background	14
2.2.1	Connectivity Classes and Connectivity Openings	14
2.2.2	Second-Generation Connectivity	15
2.2.3	Attribute Openings	18
2.3	Drawbacks of Conventional Second-Generation Connectivity	19
2.4	Mask-Based Second-Generation Connectivity	21
2.4.1	Mask-Based Connectivity	21
2.4.2	Gray-Scale Mask-Based Attribute Filters	25
2.5	Computing Second-Generation Attribute Filters	26
2.5.1	The Max-Tree Algorithm	26
2.5.2	The Dual-Input Max-Tree Algorithm	27
2.5.3	Filtering and Image Restitution	32
2.6	Experiments and Discussion	33
2.6.1	3-D Biomedical Datasets	36
2.6.2	Images of Proteins	36
2.6.3	Computational Complexity	38
2.7	Conclusions and Further Work	38

3	Filament Enhancement by Non-Linear Volumetric Filtering using Clustering-Based Connectivity	41
3.1	Introduction	41
3.2	Theory	42
3.2.1	Connectivity Classes and Openings	42
3.2.2	Clustering-Based Connectivity	44
3.3	Shape Filters	45
3.4	Experiments	46
3.5	Discussion	50
4	A Parallel Implementation of the Dual-Input Max-Tree Algorithm for Attribute Filtering	53
4.1	Introduction	53
4.2	Attribute filters	54
4.3	The Max-Tree algorithm	55
4.4	Including union-find in the Max-Tree	56
4.5	The dual-input mode	57
4.6	Concurrent merging of Max-Trees	59
4.7	Performance testing and complexity	62
4.8	Conclusions	63
5	Partition-Induced Connections and Operators for Pattern Analysis	65
5.1	Introduction	65
5.2	Theory	67
5.2.1	Connections and Connected Operators	67
5.2.2	Second-Generation Connectivity	68
5.2.3	Attribute Filters	69
5.2.4	Granulometries and Pattern Spectra	70
5.3	Partition-Induced Connections	71
5.3.1	Partitions and Connections	71
5.3.2	Countering Oversegmentation with π -Connections	72
5.3.3	π -connected Attribute Filters	73
5.3.4	Gray-scale Limitations	75
5.4	Gray-scale Pattern Analysis	76
5.4.1	Gray-Scale Pattern Spectra Using Max-Trees	76
5.4.2	Binned 2D Shape-Size Spectra	77
5.5	Diatom Identification Experiments	79
5.5.1	The ADIAC Diatom Image Database	79
5.5.2	Experimental Methods and Parametrization	79

5.5.3	The C4.5 Decision Tree Classifier	80
5.5.4	Experiments	80
5.5.5	Performance Optimization Using Combined Methods	81
5.6	Discussion of Results	83
5.7	Conclusions	85
6	Hyperconnected Attribute Filters Based on k-Flat Zones for 3D Medical Imaging	87
6.1	Introduction	87
6.2	Connections, Partitions and Operators	90
6.2.1	Connections and Partitions	90
6.2.2	Attribute Filters	91
6.2.3	Extensions to Gray-Scale	92
6.3	Hyperconnections	93
6.3.1	Hyperconnectivity Classes and Covers	93
6.3.2	Covers of k -Flat Zones	96
6.3.3	Attribute Filters Based on k -Flat Zones	98
6.4	The k -Subtractive Filtering Rule for the Max-Tree Algorithm	100
6.4.1	The Max-Tree Algorithm	100
6.4.2	The k -subtractive Implementation	101
6.5	Applications on Volumetric Data and Discussion	103
6.5.1	The foot data set	104
6.5.2	The CT-Knee data set	104
6.5.3	The MRI-Head data set	105
6.5.4	The CT-Chest data set	105
6.5.5	The spine data set	108
6.5.6	Parameter Selection and Computational Complexity	108
6.6	Conclusions	109
7	Summary	111
7.1	Conclusions	111
7.2	Future Work	113
	Samenvatting	115
	Publications	119
	Acknowledgments	121
	Bibliography	123

Chapter 1

Introduction

You don't need eyes to see, you need vision.

Faithless

IMAGE analysis is a field of computer science that has seen an increasing importance in modern days due to the commercialization of machine vision systems. The functionality of a wide range of intelligent devices, from mobile phones and simple biometric scanners to industrial automation and advanced medical imaging units, relies on the ability to make decisions based on information retrieved from pictorial data. Image enhancement and simplification, extraction of primitives and semantics for object recognition, shape modeling and representation are only a few examples of its usage that justify its crucial role.

In the plethora of tools and methods available at the disposal of image analysis packages, one can witness the almost unavoidable presence of mathematical morphology. Morphology [67, 68], which dates back to the early days of binary images, has evolved to provide robust and computationally efficient tools and operators that can handle gray-scale and color images. It is a field of computer science, based on set theory and topology [44], that finds usage in both image analysis and processing. Its solid mathematical background is easily transferred to computer programs and moreover, the simplicity of operations makes morphological algorithms an attractive alternative due to their low computational cost and memory requirements.

Most classical morphological operators decide on the intensity of a point based on information retrieved from its local neighborhood. The neighborhood is specified by the size and shape of a given structuring element (SE) and examples are dilations, erosions, structural closings and openings. Operators using a fixed structuring element cause edge distortions. This led to an effort to develop various adaptive filters such as spatially invariant morphological filters [4, 5], path-openings [1, 78], morphological amoebas [39, 40], filters using reconstruction criteria [80] and connected filters [26, 27, 29, 62, 68]. In this thesis we will be looking at the family of extensive and anti-extensive connected operators. As the name implies, these operators rely on some notion of image connectivity [62, 68] which in the case of discrete image analysis, describes the way pixels are grouped together. Assuming a binary image separated into *foreground* and *background* regions, we call a region *connected* if all its member pixels are linked through a neighborhood path, commonly established via the 4

and 8 pixel adjacency relations [17, 37]. This graph-based definition of connectivity together with other forms, are summarized in a lattice-oriented [6, 9, 69] framework termed *connected morphology*, and has a direct association to specific operators. Among the advantages of this framework is the ability to introduce various generalizations on the notion of connectivity which lead to a number of interesting operators. In this thesis three such generalizations are presented. Formalized by means of *connectivity classes*, we will be looking at the concepts of *mask-based second-generation connectivity* [57], *partition induced* or π -*connectivity* [59] and *hyperconnectivity* [58] together with their associated operators and examples of their usage.

1.1 Summary of the Work and Contributions

Connected operators [26, 27, 29, 62, 68] in the case of binary images, given a point on the image, extract a connected region marked by that point. This connected region, referred to as a *connected component*, is extracted in its entirety without edge distortions or size/shape modifications. This property is highly appreciated in many fields of modern science, like medical imaging, where object/edge modifications are simply not tolerated.

Connected components can be categorized based on several properties, such as size or shape. Early connected operators were based on reconstruction from markers [36, 88]. Markers were used to specify which connected components were to remain. Later, connected operators were developed which compute some attribute and preserve or reject components based on pre-specified criteria. These are referred to as *attribute filters* [11, 13, 49, 65] and work by comparing the measure of the component in question against a global threshold value. This is a commonly used technique for image enhancement, segmentation and object categorization. In Fig. 1.1 we see an example of vessel enhancement using a shape filter based on a non-compactness attribute. This is a rotational b-plane CT-angiogram (CTA) of the arteries of the right half of a human head, courtesy of Philips Research, Hamburg, Germany.

Vessel enhancement and segmentation is a critical task in medical imaging and there exists a number of methods for dealing with this problem [15, 18, 20–22, 38, 95–97]. Many of them however result in object distortion and suffer from low computational efficiency. Morphological attribute filters on the other hand, being shape preserving and rapid in execution, are a competitive alternative with clear advantages over a wider range of issues. In future work, a comparative study is aimed, to highlight such issues and demonstrate the potential of attribute filters in biomedical imaging. It is also possible to improve image enhancement further using combinations of attributes within the framework of vector-attribute filters [51, 84].

Apart from medical applications, connected filters have been applied to improving image compression [64, 81, 98], in video processing [63, 65], image simplification [42, 75, 76], content-based image retrieval [82], and color, vector, and multi-spectral image process-

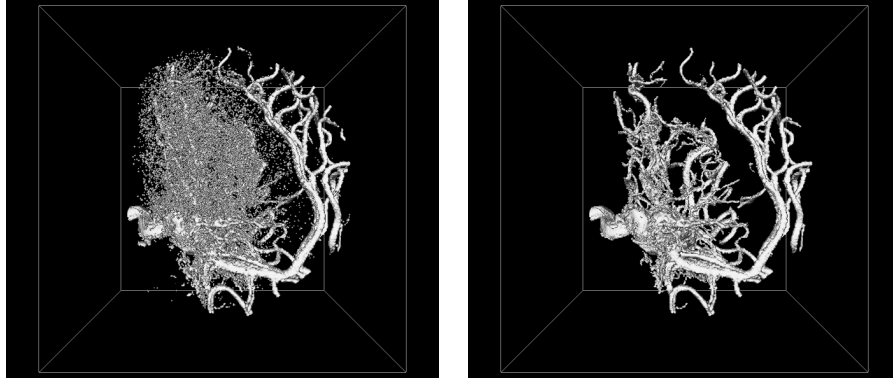


Figure 1.1: 3-D Shape filtering using 26 connectivity: The two images, starting from the left, illustrate the isosurface projections of a CTA scan containing an aneurysm and the output of an attribute filter, configured with standard connectivity, that is based on a non-compactness measure.

ing [19, 24, 75, 76]. Furthermore, connected filters have been shown to possess the desirable scale-space properties such as causality and monotonicity of the number of extrema [2, 3, 30, 34]. This extends into connectivity scale-spaces [6, 8, 74, 83] too.

Though robust enough, attribute filters configured with what we call the *standard connectivity* have certain limitations which stem from the underlying connectivity itself. The first part of this work focuses on the problem of discarded structures (failing the filter’s criterion) for which there exists empirical evidence that belong to larger objects that we aim to preserve. Using the vessels data set as before, this problem is shown in Fig.1.2 where on the left image one can see the filtered volume and on the right all the rejected objects. Note that together with the successful removal of all the noise particles, small disconnected vessel fragments that fail the filter’s criterion are also removed. Both images are shown in X-Ray rendering mode.

1.1.1 Mask-Based Second-Generation Connectivity

The standard, graph-based definition of connected components forces the filter to reject low attribute measure structures blindly, i.e., without reference to their context. One way of including this context on the filter is by looking at the distance separating them from the targeted objects. This yields a certain degree of freedom as to what we can consider connected, and is modeled by the concept of second-generation connectivity [7, 62, 69]. Operators configured with this type of connectivity can handle object clusters or contractions in the same edge preserving manner as with standard connectivity. A cluster is essentially a connected entity made of smaller components separated from each other by at most a distance r . The

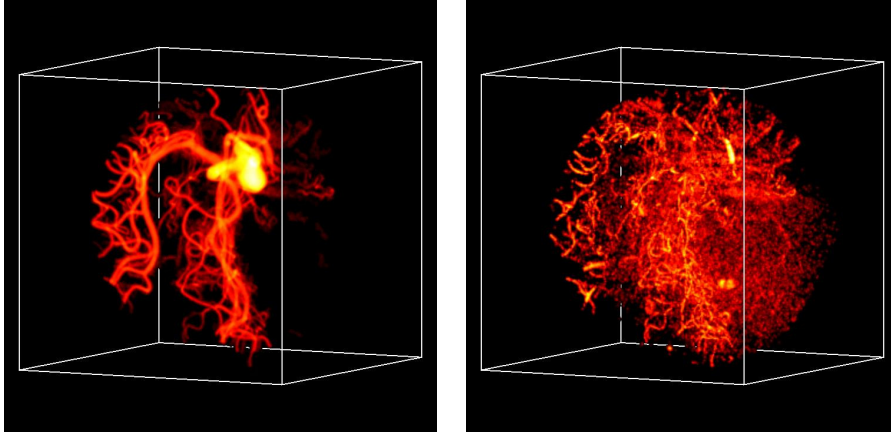


Figure 1.2: Example of missed structures using attribute filters configured with standard connectivity. On the left is the filtered volume and on the right is the set of rejected structures. Among them there exist many (vessel fragments) that we wish to preserve despite failing the attribute criterion.

parameter r is typically given by the radius of a structuring element used along with the connected operator. By contrast, object contractions are regions of connected components that are of minimum width $2r$. Referred to also as *stable components* [7], these are the invariant regions of a connected component to some anti-extensive structural operator. Prior to this work, these two types of connected entities existed separately both in theory and in algorithmics, and could only be defined from a heavily constrained set of operators. They are modeled by what is known as *clustering-based* and *contraction-based* connectivities respectively [7]. In this thesis, this is changed by presenting a unified connectivity-operator framework called "mask-based second-generation connectivity". The term "mask" describes an image, commonly a replica of the original, modified in any arbitrary way to establish a path between objects we wish to cluster together or remove thin elongated paths connecting wide object regions that we wish to break apart. Mask images may host both types of modifications or combinations of the two, leading to an unconstrained way for defining the connectivity of any given image. One of the important features of this connectivity framework is that the distance r is no longer the only parameter dictating the definition of these connected entities. Masks can be generated by taking into consideration the orientation or directionality of structures as well as any other custom criterion.

This theoretical advance is complemented by an efficient algorithm for computing attribute filters on images characterized by this new type of connectivity. It is called the *dual input Max-Tree algorithm* (DIMIT) [55] and is an extension of the original work of Salembier [65]. It builds a hierarchical image representation based on gray-scale image pairs com-

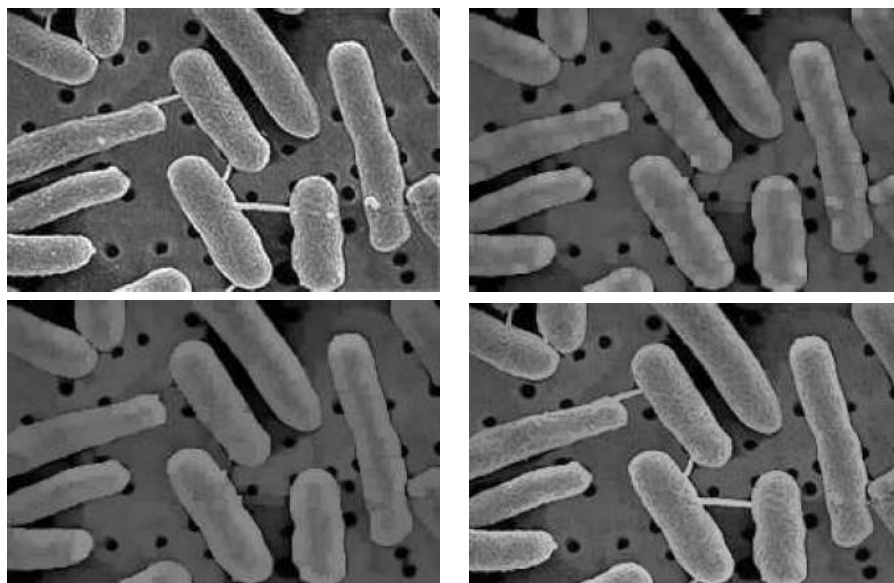


Figure 1.3: Example of over-segmentation: (left to right, top to bottom): original image; mask image obtained by a structural opening with a disc SE of radius 2; area opening using the dual input Max-Tree; area opening using the conventional Max-Tree (same area threshold).

prising the original and the mask image. In regions where the two images differ the algorithm evaluates the connectivity based on the local neighborhood properties and assigns pixels to the appropriate connected components. The algorithm handles both 2D and 3D images and provides the functionality for a wide range of different attributes to be computed. It is approximately as efficient as the original Max-Tree and supports any arbitrary mask to be used along with the original image.

A concurrent implementation is also presented to handle large 3D medical data sets which come from modern high resolution CT and MRI scanners. The parallelization strategy followed has been recently introduced for ordinary Max-Trees [93] and involves the concurrent generation and filtering of several Max-Trees, one for each thread. The algorithm uses a Union-Find type of labeling which allows for efficient merging of the trees. Speed-ups are reported and compared against the performance of concurrent implementations of ordinary Max-Trees.

1.1.2 Partition-Induced Connectivity

Operators configured with mask-based second-generation connectivity handle the clustering issue satisfactorily. Examples are given in the first two chapters where mask images are created from a variety of operators previously not supported and from classical operators such as extensive dilations and closings, respectively. The factors influencing the algorithmic performance in the case of clustering with the aid of classical operators (for mask generation) are investigated in further detail in Chapter 3. Contractions however, remain a problem in both this new framework as well as in the original.

Defining a contraction requires a mask image that is a subset of, or equal to the original. In mask-based second-generation connectivity, when talking of contractions [57], this set ordering needs not to apply on the entirety of the image. In both cases though, foreground regions of the original image that correspond to the local background in the mask, are treated as individual points, also known as *singletons*.

This re-organization of image structures has an implication on increasing attribute filters which has been investigated in [92]. In brief, an increasing attribute filter relying on a contraction-based connectivity yields the same result to that of the same filter configured with standard connectivity and operating on the contracted mask image. This holds for all cases unless the criterion has been set such that the filter is the identity operator. If that is not the case, singletons are removed indiscriminately.

In binary image processing this can be desirable and an example is given in [54]. In grayscale images though, where fine, bright details are not present in the associated masks, this causes a severe blurring effect known as *over-segmentation*. An example is shown in Fig. 1.3 where an area opening is performed on an image of bacteria. Note that in the contractive case (third image), although the bridging paths are completely removed, allowing for operators to handle each member of the bacterial colony individually, the filter output is heavily distorted.

One way of countering this issue is by organizing all path-wise connected singletons into connected components to which meaningful attributes can then be assigned. This can be modeled by an operator similar to that associated with mask-based second-generation connectivity, only requiring multiple mask images. A pilot study [54] explored some of the operator properties concluding that it is an algebraic opening thus allowing under certain conditions to establish a direct link to some form of connectivity [68]. The involvement of multiple mask images though, was thought to be restrictive and inefficient, for this an alternative formalism was investigated making use of image partitions. This offers a number of advantages, notably the direct link between connectivity classes and partitions [72], the compact expression for both connectivity classes and their associated operators, the independence of these expressions from the number of masks and the level of generalization introduced which allows for a number of other connectivities, like the standard and second-generation connectivity to be derived from this framework by imposing specific constraints.

This new framework was called *partition induced* (π) or π -connectivity [59] and is presented analytically in Chapter 5.

The organization of singletons into meaningful structures naturally leads to a redistribution of image power which in turn affects texture patterns. To explore the practical implications of this new connectivity scheme, a number of texture-based image classification experiments were carried out on two diatom image databases [12]. Pattern spectra [41, 67] computed from granulometries [11, 67, 89] were used as texture descriptors, employing π -connected operators. It is shown that these operators do not extend to gray-scale [43] trivially and a brute-force algorithm was devised for the purposes of that work. The results though obtained from limited experimentation, were conclusive enough as to what are the strengths and weaknesses of this method and what can be done to improve further the classifier's performance when using π -connected pattern spectra as feature vectors.

1.1.3 Hyperconnectivity

The first two thematic sections of this thesis deal with generalizations on the notion of connectivity. Mask-based second-generation and partition-induced connectivities are given through an interaction of mask images or partition classes with standard connectivity. In the last section, instead of having spatial modifications on the mask images dictating the form of connectivity, a new concept is investigated; that of hyperconnectivity introduced by Serra in [69].

The definition of hyperconnectivity stems from relaxing one of the conditions that describe a connectivity class. The strict set intersections are replaced with the more flexible definition of set overlap which is a controllable parameter. The idea has been put forward recently, and Braga-Neto et al. [9] showed that a large number of existing types of connectivity can be unified under one framework. The same study has also revealed certain weaknesses that emerge from this generalization which have an impact on the associated operators and little further development has been reported since. In Chapter 6 a new approach to hyperconnectivity is presented aided by a "base" connectivity. Though not general enough to address these weaknesses in their entirety, it provides operators that find use in attribute filtering based on contrast information together with structural characteristics.

Involving the notion of *k-flat zones*, the filters based on this strategy can reject high attribute measure structures that are of low contrast by controlling the parameter k (depth of a cluster of connected components along the intensity range). Similarly, they can preserve fine details that fail the filter's criterion if found in a high contrast region which meets the criterion. Fig.1.4 gives an example of the benefits offered. The data set is a C-arm X-ray scan of a phantom of a human skull, courtesy of Siemens Medical Systems, Forchheim, Germany. Regular shape filters fail to enhance the data set properly but the addition of contrast information allows the successful removal of high attribute structures which rest on

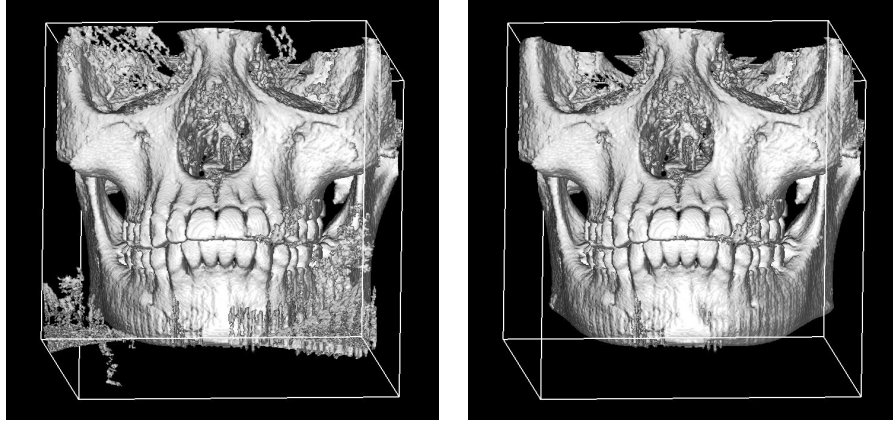


Figure 1.4: *Hyperconnected attribute filtering (left to right): the output volume of a regular filter using a sparseness criterion and the output of the same filter embedding contrast information in the form of hyperconnected sets.*

the background and are of low contrast.

The filters presented work on hyperconnected sets of maximal extent that are derived from a base connectivity class. Giving no restrictions on the nature of this base connectivity class, second-generation and other derivatives of standard connectivity may also be used. This allows for standard image representation algorithms, like the Max-Tree, to be used for their efficient computation. The method discussed is given in the form of a filtering rule such that the contrast parameter k can be changed interactively avoiding the need to recompute the tree structure of the input image/volume for each new value of k . The attribute threshold can also be changed interactively by making use of Westenberg's [90] attribute handling routines.

Experiments with non-increasing shape filters on a number of 3D medical data sets show considerable improvements on the quality of image enhancement and practically identical computational cost when compared to regular filtering.

1.2 Thesis Organization

This thesis is organized in five core chapters which are all journal or conference papers published or submitted for publication. To make each chapter as self contained as possible, some of the concepts presented are repeated in other chapters too.

In Chapter 2 the notion of connectivity, connected operators and attribute filters is presented. The notion of second-generation connectivity given through the axiomatic defini-

tion of clustering based and partitioning based connectivity is then presented followed by a study on the drawbacks and limitations of the existing formalism. The idea of mask-based second-generation connectivity, its operators and how they extend to gray-scale follows and is complemented by the introduction of the dual-input Max-Tree algorithm for which a detailed explanation is given. Experiments to demonstrate the usability of this new framework conclude this chapter together with a short discussion on our findings.

In Chapter 3, using the DIMT algorithm we experiment on the factors influencing the quality and performance of shape filters on 3D medical data sets. The work aims at pinpointing the critical filter parameters and how they influence the filter output. The experiments refer to the purely clustering case only. Moreover, a new method for computing attributes is presented which minimizes a problem reported in Chapter 2; that of noise clustering.

Chapter 4, presents a concurrent implementation of the DIMT algorithm that is based on the parallelizing strategy followed for regular Max-Trees. The challenging part of this work is in the merging process of the individual tree structures that emerge from each thread. Clusters and contractions co-existing on 3D volume sets are brought together in an innovative way which is described in detail.

The fifth chapter, moves on to the notion of π -connectivity. It starts off with an introduction on the basic connectivity concepts and gives some brief background theory on granulometries and pattern spectra. Based on the work of Serra on image partitions, the concept of π -connectivity and its associated operators are then presented together with a small section giving reasoning why such operators cannot extend to gray-scale trivially. To evaluate their performance in practical applications, the operators are used to define a type of pattern spectrum-like structure which is used in a texture-based image classification problem. Due to the limitations discussed in gray-scale extensions we devise a brute-force algorithm for computing this structure, which though not very efficient bypasses these limitations. Using two diatom image data bases and the C4.5 decision tree classifier, we revisit the problem of diatom classification, subject of the ADIAC [12] project, and draw our conclusions on the performance reported.

Chapter 6 deals with the concept of hyperconnectivity based on k -flat zones. The introduction clarifies the differences between k - and λ -flat zones with the aid of an example. After a short briefing on connectivity concepts, the notion of hyperconnectivity and covers is presented. Covers which are similar to image partitions, only allowing for set overlap, if made up of sets of k -flat zones lead to an explicit formalization of hyperconnectivity whose associate operators can capture contrast information. A hyperconnected attribute filter of such kind is introduced relying on a base connectivity class. Implementing such filters in the form of a filtering rule is the next topic discussed, and a pseudo code is given. Experiments on 3D medical data sets and a discussion of our findings conclude the chapter.

The thesis concludes with a summary of this work and all the findings, followed by the bibliographical listing.

Chapter 2

Mask-Based Second-Generation Connectivity and Attribute Filters

I may not have gone where I intended to go, but I think I have ended up where I needed to be.

Douglas Adams

Abstract

Connected filters are edge-preserving morphological operators, which rely on a notion of connectivity. This is usually the standard 4- and 8-connectivity, which is often too rigid since it cannot model generalized groupings such as object clusters or partitions. In the set-theoretical framework of connectivity these groupings are modeled by the more general second-generation connectivity. In this paper we present both an extension of this theory, and provide an efficient algorithm based on the Max-Tree to compute attribute filters based on these connectivities. We first look into the drawbacks of the existing framework that separates clustering and partitioning and is directly dependent on the properties of a pre-selected operator. We then propose a new type of second-generation connectivity termed mask-based connectivity which eliminates all previous dependencies and extends the ways the image domain can be connected. A previously developed Dual-Input Max-Tree algorithm for area openings is adapted for the wider class of attribute filters on images characterized by second-generation connectivity. CPU-times for the new algorithm are comparable to the original algorithm, typically deviating less than 10% either way.

2.1 Introduction

IN discrete image analysis the set-theoretic concept of connectivity [68] describes the way pixels are grouped to form connected components or flat-zones in gray-scale [66]. Connected components are image regions of constant intensity in which pixels are characterized by a path-wise connectedness relation. Typically on the two-dimensional (2-D) discrete space \mathbb{Z}^2 sets of pixels are either 4- or 8-connected [17, 37].

Based on the notion of connectivity, a family of morphological operators [26] known as *connected filters* [29, 66] has been developed which interact with the connected components rather than individual pixels. This prevents edge distortion, a property highly desirable in many applications. Connected components can either be removed or remain intact but new ones cannot emerge. Early members of this family were openings by reconstruction, for which efficient algorithms have been developed [88]. Furthermore, the concept of attribute filters [11] was introduced, which allows filtering based on the connected component attributes. Examples of this are attribute openings, closings, thinnings and thickenings [11, 13, 29, 86].

In recent years several theoretical developments concerning generalizations of the notion of connectivity have been presented [9], aiming to improve the robustness and increase the versatility of these filters. These generalizations aim at modeling object clusters and partitions in an edge preserving manner. A well established approach known as *second-generation connectivity* [7, 62, 69] handles both cases independently by creating a "child" connectivity class, by using some operator. Second-generation connectivity can be classified as either *clustering* or *contraction*-based [7] depending on whether the operator expands, or contracts the original image. An example of filtering based on clustering connectivity is given in Fig. 2.1 illustrating an Anabaena complex. Assuming we target the largest complex, using standard 4- and 8-connectivity we can only retrieve the bigger fragment of the two, which are separated by the heterocyst - bottom left image. Instead, if the connected filter is defined on the more general clustering-based connectivity, the two fragments merge as illustrated in top right image, and the filter considers them as one object - bottom right image. The original image has been obtained from <http://www.f-suiki.or.jp/suisitu/saikin/saikin.htm>.

Second-generation connectivity is realized by means of a connectivity opening which is associated with a structural operator. The dependency on this operator imposes constraints as to how the image domain can be connected, and apart from clustering and partitioning, no further cases such as combinations of the two are supported. In our work we counter this limitation by introducing a composite connectivity opening in which all dependencies to the structural operator are eliminated. Instead, we propose an association with a *connectivity mask* which is an image containing some arbitrary transformation of the original. This yields a single framework termed *mask-based connectivity* that accounts for all possible ways the image domain can be connected. This includes the two known cases of clustering and partitioning through the design of filters which yield results identical to the previous approach, even though formally based on different connectivity classes. The difference between these classes lies in the fact that, for a given operator, we generate a different connectivity class for each target image, rather than one, generally applicable connectivity class.

Algorithmic realizations of second-generation connectivity originally suggested the use of binary and gray-scale reconstruction operators for recovering the object clusters or par-

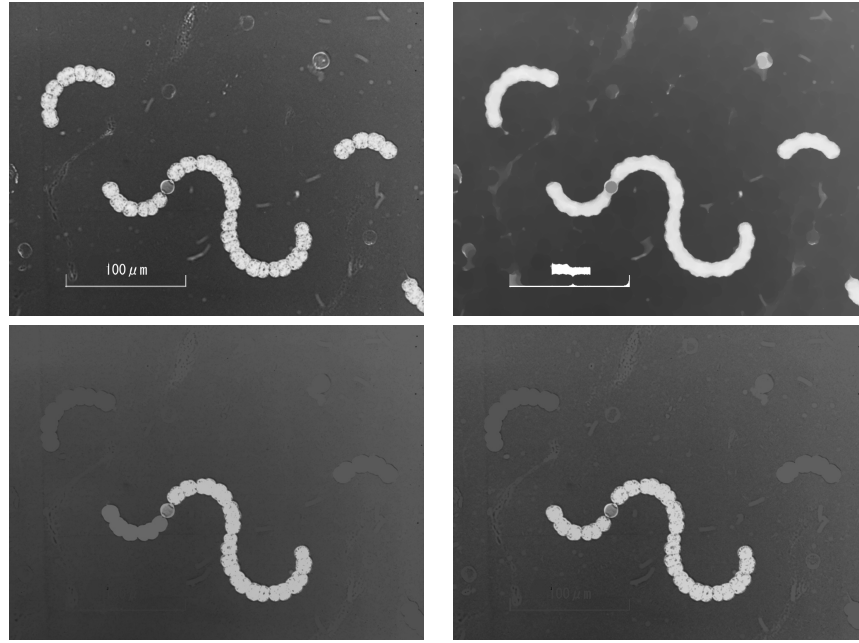


Figure 2.1: Area opening using clustering-based connectivity: original image (top left); the expanded set obtained by a structural closing (top right); the filtered image using an area criterion relying on the standard 4-connectivity (bottom left) and clustering-based connectivity (bottom right).

titions [7]. This introduced a family of filters based on width as the attribute criterion. An efficient algorithm for gray-scale area filters using second-generation connectivity has also been presented [55]. This method builds a hierarchical image representation based on gray-scale image pairs comprising the original image and its modified replica. In regions where the two images differ the algorithm evaluates the connectivity based on the properties of the structural operator and assigns pixels to the appropriate connected components. The algorithm which is inspired by Salembier et al. [65], is referred to as the Dual-Input Max-Tree algorithm [55] and supports both clustering and partitioning. In this work we employ the Dual-Input Max-Tree modified for the more general family of gray-scale attribute filters on 2-D and 3-D datasets characterized by mask-based second-generation connectivity. We demonstrate its capacity on biomedical images and 3-D datasets using non-increasing shape filters based on moment invariants, and provide the functionality to extend to other filter types.

The structure of this paper is organized as follows: in Section 2.2 a number of preliminaries is presented. These are the fundamental concepts of connectivity classes and connec-

tivity openings, the notion of second-generation connectivity, and attribute filters. In Section 2.3 we investigate the drawbacks of the existing second-generation connectivity framework. Section 2.4 introduces the mask-based connectivity scheme and formalizes an expression of attribute openings associated to it. Following this, Section 2.5 gives a short introduction on the Max-Tree structure complemented by the description of the Dual-Input Max-Tree algorithm adopted for mask-based connectivity representation. Section 2.6 gives a number of examples on attribute filtering and a brief discussion on the results while conclusions are summarized in Section 2.7.

2.2 Theoretical Background

2.2.1 Connectivity Classes and Connectivity Openings

This section briefly outlines the concept of connectivity from the set-oriented morphological perspective. For the purpose of this analysis we assume a universal (non-empty) set E and denote by $\mathcal{P}(E)$ the collection of all subsets of E .

Definition 1. A family $\mathcal{C} \subseteq \mathcal{P}(E)$ for any arbitrary set E , is called a connectivity class if it satisfies:

1. $\emptyset \in \mathcal{C}$ and for all $x \in E$, $\{x\} \in \mathcal{C}$,
2. for any $\{A_i\} \subseteq \mathcal{C}$ for which $\bigcap A_i \neq \emptyset \Rightarrow \bigcup A_i \in \mathcal{C}$

This means that both the empty set and singleton sets are connected, and any union of sets which have a non-empty intersection is also connected. Members of \mathcal{C} are called *connected sets* [68–70]. The family of all singleton sets is denoted by $\mathcal{S} \subseteq \mathcal{C}$.

Addressing connected regions in binary images is often more practical by means of *connected components* or *grains* C [29]. If C is a grain of X we denote this $C \in X$. A connected component C of a binary image X is a connected set of maximal extent, in the sense that there is no set $C' \supset C$ such that $C' \subseteq X$ and $C' \in \mathcal{C}$.

Connected components are accessed by means of a *connectivity opening* Γ_x which is an operator extracting the union of all connected sets within X that have a point $x \in E$ in their intersection, i.e.

$$\Gamma_x(X) = \bigcup \{A_i \in \mathcal{C} \mid x \in A_i, A_i \subseteq X\} \quad (2.1)$$

for every $X \subseteq E$. From (2.1) it is trivial to show that Γ_x is an algebraic opening [61] marked by x , i.e. it is an increasing, anti-extensive and idempotent operator. Furthermore $\forall x \notin X$, $\Gamma_x(X) = \emptyset$.

Evidently connectivity classes and connectivity openings are inter-related. This is formally given by the following theorem [62, 68, 69].

Theorem 1. *The datum of a connectivity class \mathcal{C} in $\mathcal{P}(E)$ is equivalent to the family $\{\Gamma_x \mid x \in E\}$ of openings on x such that:*

1. Γ_x is an algebraic opening marked by $x \in E$
2. for all $x \in E$, we have $\Gamma_x(\{x\}) = \{x\}$,
3. for all $X \subseteq E$, and all $x \in E$, we have $x \notin X \Rightarrow \Gamma_x(X) = \emptyset$,
4. for all $X \subseteq E$, $x, y \in E$, if $\Gamma_x(X) \cap \Gamma_y(X) \neq \emptyset \Rightarrow \Gamma_x(X) = \Gamma_y(X)$, i.e. $\Gamma_x(X)$ and $\Gamma_y(X)$ are equal or disjoint.

This shows that connectivity openings characterize uniquely the connectivity class with which they are associated and that there is a one-to-one correspondence between the two. We see this from two points of view [62]:

(a) From the connectivity class to the system of connectivity openings: Γ_x is the union of all sets A in the connectivity class \mathcal{C} , such that $x \in A$ and $A \subseteq X$, and

(b) from the system of connectivity openings to the connectivity class \mathcal{C} : the connectivity class \mathcal{C} is formed of all $\Gamma_x(X)$ for $x \in E$ and $X \subseteq E$.

Concluding we see that to prove a family of sets is a connectivity class, it is sufficient to show that the operator extracting these sets is a connectivity opening satisfying the four conditions of Theorem 1.

2.2.2 Second-Generation Connectivity

Given a connectivity class \mathcal{C} it is possible to generate a *child* class with either reduced or enriched members by modifying its associated connectivity opening. This is referred to as *second-generation connectivity* [62,69,70] and aims at modelling object clusters or partitions that cannot be captured otherwise.

Each of the two cases is defined separately and we identify second-generation connectivity as either *clustering* or *contraction*-based [7]. In both cases the connectivity openings Γ_x^ψ of the family associated to the child connectivity class are given in an expression dependent on a structural operator ψ such as a dilation, a closing or an opening.

Clustering-Based Connectivity

The first case of second-generation connectivity describes groups of image objects that can be perceived as clusters of connected components if their relative distances are below a given threshold. This is controlled by the size of the structuring element used along with an operator ψ termed *clustering* [7, 62, 69]. Following is a list summarizing the properties required to define a clustering:

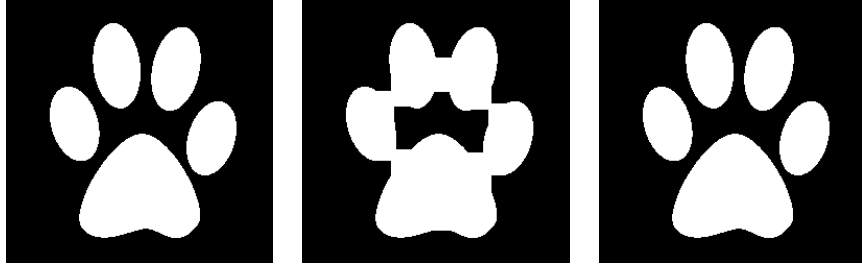


Figure 2.2: Clustering Sets: The original image X (left) illustrates five separate objects which expanded by ψ yield the sets making up the cluster (middle). By intersecting the connected components of $\psi(X)$ with X , the operator $\Gamma_x^\psi(X)$ extracts the cluster of the previously disconnected objects (right).

1. ψ is increasing and extensive.
2. $\psi(\mathcal{C}) \subseteq \mathcal{C}$.
3. For a family $\{X_i\}$ in $\mathcal{P}(E)$ such that $\psi(X_i) \in \mathcal{C}$, $\forall i$, and $\bigcap_i X_i \neq \emptyset \Rightarrow \psi(\bigcup X_i) \in \mathcal{C}$.
4. ψ does not create connected components; i.e., if $\forall x \in C$ and $C \in \psi(X) \Rightarrow X \cap C \neq \emptyset$.
5. ψ treats the clusters of X independently; i.e., if $\forall x \in C$ and $C \in \psi(X) \Rightarrow \psi(X \cap C) = C$.

Further analysis of each item is given in [7].

If the operator ψ satisfies the first three properties, it is referred to as a *weak clustering* or simply clustering.

Definition 2. Let \mathcal{C} be a connectivity class in $\mathcal{P}(E)$ and ψ be a clustering operator on $\mathcal{P}(E)$. Then

$$\mathcal{C}^\psi = \{X \in \mathcal{P}(E) \mid \psi(X) \in \mathcal{C}\} \quad (2.2)$$

is a clustering-based connectivity class with

$$\mathcal{C} \subseteq \mathcal{C}^\psi. \quad (2.3)$$

Operator ψ is a *strong clustering* if, and only if it satisfies all five properties above. Typical examples of clustering operators are certain extensive dilations and closings, using connected structuring elements.



Figure 2.3: Partitioning Sets: the original image containing a single connected component (left), the set of stable components given by $\psi(X)$ (middle), an independent connected component previously connected to the grid (right).

Definition 3. Let $\{\Gamma_x \mid x \in E\}$ be the connectivity openings associated with \mathcal{C} . If ψ is a strong clustering on $\mathcal{P}(E)$, the family of connectivity openings $\{\Gamma_x^\psi \mid x \in E\}$ associated to \mathcal{C}^ψ are given by

$$\Gamma_x^\psi(X) = \begin{cases} \Gamma_x(\psi(X)) \cap X, & \text{if } x \in X \\ \emptyset, & \text{otherwise} \end{cases} \quad (2.4a)$$

$$(2.4b)$$

An example of clustering sets is illustrated in Fig. 2.2 where the first image from the left shows a set of five individual sets connected components and the second, an expanded replica by a structural closing. The connectivity opening of (2.4) for $(x, y) = (128, 200)$ extracts the cluster of the connected components as illustrated in the third image.

Contraction-Based Connectivity

The second case is a partitioning scheme in which wide object regions bridged in the original image by narrow elongated structures can be treated as separate objects [7, 61, 62]. The "narrowness" of these structures is determined by the size the structuring element used along with an increasing and anti-extensive operator ψ on $\mathcal{P}(E)$, called a *contraction*. Furthermore, any set $X \subseteq E$ which is invariant to ψ is called *stable*, i.e. $\psi(X) = X$.

Restricting the original connectivity class \mathcal{C} by turning all connected members that are not invariant to ψ to connected singleton sets, yields a child connectivity class defined as follows:

Definition 4. Let \mathcal{C} be a connectivity class in $\mathcal{P}(E)$ and ψ be a contraction on $\mathcal{P}(E)$. Then

$$\mathcal{C}^\psi = \{\emptyset\} \cup \mathcal{S} \cup \{X \in \mathcal{C} \mid \psi(X) = X\} \quad (2.5)$$

is a contraction-based connectivity class with

$$\mathcal{C}^\psi \subseteq \mathcal{C}. \quad (2.6)$$

Contractions are typically structural openings. A necessary condition to define the family of connectivity openings associated with \mathcal{C}^ψ is for ψ to be locally invariant with respect to \mathcal{C} for any $X \subseteq E$ [7], i.e.:

$$\psi(X) = X \Rightarrow \psi(\Gamma_x(X)) = \Gamma_x(X), \forall x \in E. \quad (2.7)$$

This means that for any set X invariant to ψ , all connected components of X must also be invariant to ψ . An example of a locally invariant opening is a structural opening with a connected structuring element.

Definition 5. Let $\{\Gamma_x \mid x \in E\}$ be the connectivity openings associated with \mathcal{C} . If ψ is an opening on $\mathcal{P}(E)$ locally invariant with respect to \mathcal{C} , the family of connectivity openings $\{\Gamma_x^\psi \mid x \in E\}$ associated to \mathcal{C}^ψ are given by

$$\Gamma_x^\psi(X) = \begin{cases} \Gamma_x(\psi(X)) & \text{if } x \in \psi(X) \\ \{x\} & \text{if } x \in X \setminus \psi(X) \\ \emptyset & \text{otherwise} \end{cases} \quad \begin{array}{l} (2.8a) \\ (2.8b) \\ (2.8c) \end{array}$$

Partitioning an image with this scheme does not modify the existing edges and the union of all stable sets with the singletons that complement them yields back the original image. An example is illustrated in Fig. 2.3 where there exists a single connected component which we supposedly would like to handle as five separate objects disconnected from the grid. The grid in this example represents a background object connecting the objects of interest in a non-desirable way. Applying a structural opening ψ on X removes the grid and yields the set of all the components invariant to ψ , termed *stable* by [7] (middle image). Elements of the grid removed by ψ are treated as singletons in \mathcal{C}^ψ . Applying $\Gamma_x^\psi(X)$ will extract each of the five objects seen in the middle image separately (last image for $(x, y) = (128, 200)$).

2.2.3 Attribute Openings

Binary attribute openings [11] are a family of connected filters [29, 66] that incorporate a trivial opening Γ_Λ on the output of a connectivity opening Γ_x . The trivial opening accepts or rejects connected components subject to an increasing and shift invariant criterion Λ given by:

$$\Lambda(C) = (\text{Attr}(C) \geq \lambda) \quad (2.9)$$

with $\text{Attr}(C)$ some real-value attribute of the connected component C , and λ an attribute threshold. The increasingness of Λ implies that if a set A satisfies Λ then any set B such that $B \supseteq A$ satisfies Λ as well. We can summarize the definition of Γ_Λ to the following $\Gamma_\Lambda : \mathcal{C} \rightarrow \mathcal{C}$ operating on $C \in \mathcal{C}$ yields C if $\Lambda(C)$ is true, and \emptyset otherwise. Furthermore, $\Gamma_\Lambda(\emptyset) = \emptyset$. The binary attribute opening can be defined as follows:

Definition 6. The binary attribute opening Γ^Λ of a set X with increasing criterion Λ is given by:

$$\Gamma^\Lambda(X) = \bigcup_{x \in X} \Gamma_\Lambda(\Gamma_x(X)) \quad (2.10)$$

An example is the area opening [87]. Note that if Λ is non-increasing we have an *attribute thinning* [11] or *non-increasing grain filter* [29] rather than an attribute opening. An example is the scale-invariant elongation criterion in which $Attr(C) = I(C)/A^2(C)$, with $I(C)$ the moment of inertia of C and $A(C)$ the area [85, 86, 94]. For the volume set that we demonstrate later on, the moment of inertia is given by

$$I(C) = \frac{V(C)}{4} + \sum_{\mathbf{x} \in C} (\mathbf{x} - \bar{\mathbf{x}})^2 \quad (2.11)$$

in which $V(C)$ denotes the volume of C , and the elongation (or non-compactness) [94] is measured by the ratio

$$Attr(C) = \frac{I(C)}{V^{5/3}(C)}. \quad (2.12)$$

$Attr(C)$ has a minimum for a sphere (0.23) and increases rapidly with elongation.

2.3 Drawbacks of Conventional Second-Generation Connectivity

This section summarizes some drawbacks of second-generation connectivity due to the dependency of the associated connectivity openings on the properties of the structural operators used along with them. The objective is to demonstrate that there exist useful structural operators that are excluded because they do not comply with the requirements listed in Section 2.2.2.

The connectivity openings associated with the two cases of second-generation connectivity reviewed in section 2.2.2 can be summarized in a unified formalism given by:

$$\Gamma_x^\psi(X) = \begin{cases} \Gamma_x(\psi(X)) \cap X & \text{if } x \in X \cap \psi(X) & (2.13a) \\ \{x\} & \text{if } x \in X \setminus \psi(X) & (2.13b) \\ \emptyset & \text{otherwise} & (2.13c) \end{cases}$$

It is evident that if ψ is a strong clustering (extensive) the situation in which an element $x \in X \setminus \psi(X)$ cannot occur therefore (2.13) simplifies to (2.4). If ψ is a contraction (anti-extensive) then for $x \in X \cap \psi(X)$ the corresponding term in (2.13) simplifies to $\Gamma_x(\psi(X))$ due to the anti-extensivity of ψ and hence we obtain (2.8).

Merging the two cases of clustering and partitioning as operations in a single expression unifies implementations of second-generation connectivity like in [55]. The connectivity

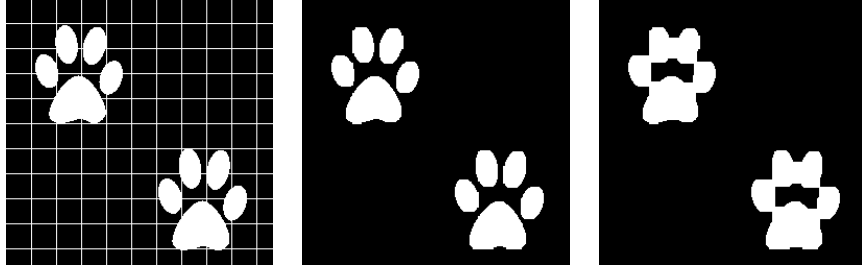


Figure 2.4: Clustering Partitioned Sets: (a) original image X , (b) the contracted set by ψ_p , and (c) $\psi_{cp}(X)$: expanding the contractions by ψ_c on $\psi_p(X)$.

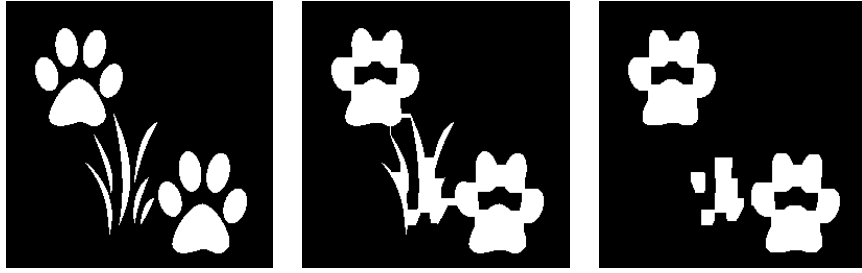


Figure 2.5: Partitioning Clustered Sets: (a) original image X , (b) the expanded set by ψ_c and (c) $\psi_{pc}(X)$: the contraction by ψ_p on the expanded set $\psi_c(X)$.

opening of (2.13) also allows a number of other structural operators to be used, but because they violate the properties described in section 2.2.2 they prevent obtaining a valid connectivity class.

Typical examples are the *alternating-sequential filters* or *AS-filters* [27, 28, 68] which are excluded since they are neither extensive nor anti-extensive operators. AS-filters modify the original image by introducing regions that appear as the result of local clustering or partitioning. They are defined as either:

$$\psi_{cp} = \psi_c \psi_p \quad (2.14)$$

or

$$\psi_{pc} = \psi_p \psi_c \quad (2.15)$$

where ψ_c and ψ_p are closings and openings respectively by a connected structuring element. Examples of each case are illustrated in Fig. 2.4 and Fig. 2.5.

When clustering contracted sets, first the input image is contracted to disconnect objects from thin elongated structures like the grid in the first image of Fig. 2.4 which is no longer

present in the second. If the resulting objects are to be treated as groupings a further clustering operator is applied which merges the neighboring connected components to the desired clusters as in the right image. By contrast, when partitioning expanded sets we aim at disconnecting the expanded objects as in the middle image of Fig. 2.5 from unwanted narrow structures by applying a contracting operator. The grid as in Fig. 2.3 represents a background object.

An example of operators violating the increasingness property are directional Minkowski additions which perform a maximum operation along the direction of elongation using adaptive structural elements. They are non-increasing extensive operators hence for $X \subseteq Y$ we may obtain $\psi(X) \not\subseteq \psi(Y)$ (see Fig. 2.6). For an element x with $x \in X \cap \psi(X)$, $x \notin Y \cap \psi(Y)$ therefore $\Gamma_x^\psi(X) \not\subseteq \Gamma_x^\psi(Y)$. This implies that Γ_x^ψ is non-increasing hence not an algebraic opening.

A typical case of an increasing, anti-extensive and non-idempotent operator is that of erosions. Erosions violate the idempotence of the contraction-based connectivity opening which requires that

$$\Gamma_x^\psi(\Gamma_x^\psi(X)) = \Gamma_x^\psi(X). \quad (2.16)$$

For the case that $x \in X \cap \psi(X)$ according to (2.13) this is:

$$\Gamma_x^\psi(\Gamma_x^\psi(X)) = \Gamma_x^\psi(\Gamma_x(\psi(X)) \cap X) \quad (2.17)$$

This is equivalent to $\Gamma_x^\psi(\Gamma_x(\psi(X)))$ due to the anti-extensivity of ψ and can be written as:

$$\Gamma_x^\psi(\Gamma_x^\psi(X)) = \Gamma_x(\psi(\Gamma_x(\psi(X)))) \neq \Gamma_x(\psi(X)) \quad (2.18)$$

i.e. erosions are not valid operators.

All three cases demonstrate incompatibility with the existing second-generation connectivity framework by violating the properties of the connectivity opening Γ_x^ψ through the properties of ψ . In Section 2.6.2 we show practical examples of the first two cases.

2.4 Mask-Based Second-Generation Connectivity

2.4.1 Mask-Based Connectivity

The limitations on the nature of the second-generation connectivity to clustering or partitioning only can be eliminated if the associated family of connectivity openings is no longer dependant on a structural operator. For this purpose we present an alternative scheme in which we associate the connectivity openings to the resulting image of some arbitrary transformation on X . We call this the *connectivity mask* and since we initially make no assumptions as to how it is created, we denote it as a generic set $M \subseteq E$. The key notion is that we only

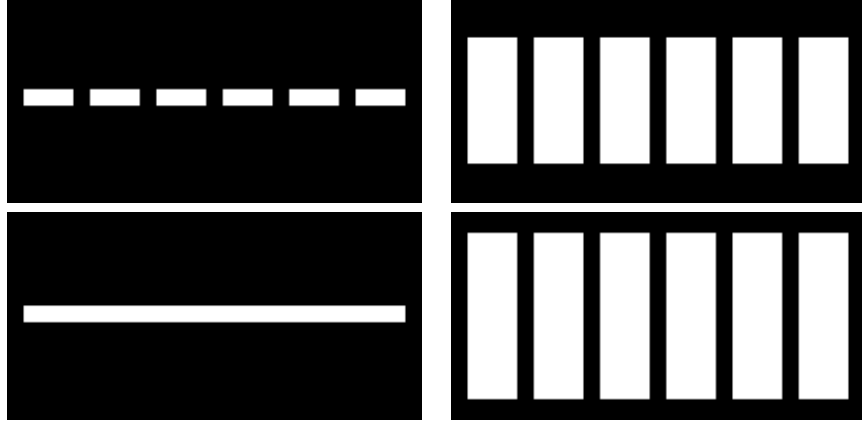


Figure 2.6: An example of an extensive and non-increasing operator (adaptive Minkowski addition) that violates increasingness of the connected openings; original image X (top left), image $Y \supseteq X$ (top right), $\psi(X)$ (bottom left) and $\psi(Y)$ (bottom right). Note that the connected components are expanded along the direction of elongation.

apply the modifying operator ψ to image X once to obtain mask M , whereas the operator-based framework applies ψ each time a filter is applied which uses connectivity based on \mathcal{C}^ψ . Thus, if we were to compute a granulometry based on attribute filters [11] within this framework, mask M would be precomputed, and all subsequent openings performed using the same M . Based on M , the connectivity openings $\{\Gamma_x^M \mid x \in E\}$ "mask" the desired members of \mathcal{C} to the child class \mathcal{C}^M . Apart from the family of the singleton sets and the empty set which are essential in the definition of a connectivity class, the members of \mathcal{C}^M can be summarized to all subsets A of the universal set E that are included in some grain of M , denoted as $\Gamma_x(M)$. This is formalized accordingly:

Definition 7. Let $\mathcal{C} \subseteq \mathcal{P}(E)$ be a connectivity class and $M \subseteq E$ a connectivity mask for an image X . The mask-based second-generation connectivity class \mathcal{C}^M is given by:

$$\mathcal{C}^M = \{\emptyset\} \cup \mathcal{S} \cup \{A \subseteq E \mid \exists x \in E : A \subseteq \Gamma_x(M)\} \quad (2.19)$$

Inspired by (2.13) we propose an association of \mathcal{C}^M with a family of connectivity openings $\{\Gamma_x^M \mid x \in E\}$ as follows.

Proposition 1. Let \mathcal{C} be a connectivity class in $\mathcal{P}(E)$ and $X, M \subseteq \mathcal{C}$ be the original image and the connectivity mask respectively.

1. Then the operator

$$\Gamma_x^M(X) = \begin{cases} \Gamma_x(M) \cap X & \text{if } x \in X \cap M & (2.20a) \\ \{x\} & \text{if } x \in X \setminus M & (2.20b) \\ \emptyset & \text{otherwise} & (2.20c) \end{cases}$$

extracting subsets of X found within the grains of M is a connectivity opening, and

2. the family $\{\Gamma_x^M \mid x \in E\}$ is associated to C^M .

Proof. (1) To prove this proposition we must show that $\Gamma_x^M(X)$ meets the requirements of Theorem 1. First we show that it is an algebraic opening, i.e. it is an anti-extensive, increasing and idempotent operator.

Anti-extensivity is trivial since for all three cases of (2.20) $\Gamma_x^M(X) \subseteq X$. Increasingness requires if $X \subseteq Y \Rightarrow \Gamma_x^M(X) \subseteq \Gamma_x^M(Y)$. We can identify two important cases: (i) $x \notin X$ and (ii) $x \in X$. In the first case $\Gamma_x^M(X) = \emptyset$ so whichever case of (2.20) holds for $\Gamma_x^M(Y)$ we have $\Gamma_x^M(X) \subseteq \Gamma_x^M(Y)$. In the second case $x \in Y$ because $X \subseteq Y$. Again we identify two cases: (i) $x \in M$ and (ii) $x \notin M$ corresponding to (2.20a) and (2.20b), respectively. If $x \in M$ we have

$$\Gamma_x^M(X) = \Gamma_x(M) \cap X \quad (2.21)$$

and

$$\Gamma_x^M(Y) = \Gamma_x(M) \cap Y. \quad (2.22)$$

Obviously, if $X \subseteq Y$ then $\Gamma_x(M) \cap X \subseteq \Gamma_x(M) \cap Y$ and therefore we have $\Gamma_x^M(X) \subseteq \Gamma_x^M(Y)$. Finally, if $x \notin M$ we have

$$\Gamma_x^M(X) = \Gamma_x^M(Y) = \{x\}, \quad (2.23)$$

so that $\Gamma_x^M(X) \subseteq \Gamma_x^M(Y)$ holds in all three cases of (2.20). For idempotence we require that

$$\Gamma_x^M(\Gamma_x^M(X)) = \Gamma_x^M(X). \quad (2.24)$$

Again we treat the three cases of (2.20) separately. The simplest is the case (2.20c), in which $\Gamma_x^M(X) = \emptyset$. Because of anti-extensivity $\Gamma_x^M(\emptyset) = \emptyset$, so in this case (2.24) holds. In the case of (2.20b) we have $\Gamma_x^M(X) = \{x\}$. Obviously $x \in \{x\} \setminus M$, so that in $\Gamma_x^M(\{x\})$ the case of (2.20b) applies again, and (2.24) holds. Finally, if $x \in X \cap M$ we have

$$\Gamma_x^M(X) = \Gamma_x(M) \cap X. \quad (2.25)$$

In this case $x \in \Gamma_x^M(X) \cap M$, so (2.20a) applies:

$$\begin{aligned} \Gamma_x^M(\Gamma_x^M(X)) &= \Gamma_x(M) \cap \Gamma_x^M(X) \\ &= \Gamma_x(M) \cap \Gamma_x(M) \cap X \\ &= \Gamma_x(M) \cap X = \Gamma_x^M(X), \end{aligned} \quad (2.26)$$

and therefore idempotence holds in all three cases. Note that no restriction is placed on M , i.e. $\Gamma_x^M(X)$ is an algebraic opening for any $M \subseteq E$.

The second requirement of Theorem 1 states that $\Gamma_x^M(\{x\}) = \{x\}, \forall x \in E$. In the case where $x \in X \cap M$, $\Gamma_x^M(\{x\}) = \Gamma_x(\{x\}) \cap M = \{x\}$ for whatever M . Similarly if $x = \{x\} \setminus M$, $\Gamma_x^M(\{x\}) = \{x\}$ from (2.20b). The third requirement states that $\forall X \subseteq E$, and $\forall x \in E$, if $x \notin X \Rightarrow \Gamma_x(X) = \emptyset$. From (2.20c) we see that if $x \notin X \Rightarrow x \in \emptyset$ therefore $\Gamma_x^M(X) = \emptyset$.

To prove the fourth requirement of Theorem 1 we require that for any $x, y \in E$, the connected components returned by the connectivity opening which are marked by x and y are either equal or disjoint, i.e. $\Gamma_x^M(X) \cap \Gamma_y^M(X) = \emptyset$ or $\Gamma_x^M(X) = \Gamma_y^M(X)$. We identify four cases:

1. If $x, y \in X \cap M$, then $\Gamma_x^M(X)$ and $\Gamma_y^M(X)$ are equal or disjoint, because $\Gamma_x(M)$ and $\Gamma_y(M)$ are equal or disjoint by the definition of connectivity openings.
2. If $x, y \in X \setminus M$, then $\Gamma_x^M(X) = \{x\}$ and $\Gamma_y^M(X) = \{y\}$, which are equal or disjoint.
3. If x or $y \notin X$, then $\Gamma_x^M(X) \cap \Gamma_y^M(X) = \emptyset$.
4. If $x \in X \cap M$ and $y \in X \setminus M$, then $\Gamma_x^M(X) \cap \Gamma_y^M(X) = \emptyset$ because $(X \cap M) \cap (X \setminus M) = \emptyset$, i.e. the two connected components are defined over disjoint partitions of X .

(2) So far we showed that $\Gamma_x^M(X)$ satisfies Theorem 1 hence the family $\{\Gamma_x^M \mid x \in E\}$ is associated with a connectivity class. Now we verify that this is \mathcal{C}^M according to Definition 7.

A connectivity class is equivalent to the union over all $x \in E$ of the invariance domains of the associated connectivity openings [29, 62]. The invariance domain of Γ_x^M contains besides the empty set, all connected sets in \mathcal{C}^M that contain x , i.e.:

$$\text{Inv}(\Gamma_x^M) = \{\emptyset\} \cup \{\{x\}\} \cup \{A \subseteq E \mid x \in A, A \subseteq \Gamma_x(M)\}, \quad (2.27)$$

for each $x \in E$. By $\{\{x\}\}$ we denote the set containing the singleton set $\{x\}$, to distinguish the two. The first term is trivial since the empty set is invariant to every Γ_x^M , i.e. $\Gamma_x^M(\emptyset) = \emptyset$. The second is included because Γ_x^M is a connectivity opening so that $\Gamma_x^M(\{x\}) = \{x\}$ for all $x \in E$. The last term states that $\Gamma_x^M(A) = A$ if $x \in A$ and $A \subseteq \Gamma_x(M)$, which follows from (2.20a). This readily shows that $\text{Inv}(\Gamma_x^M) \subseteq \mathcal{C}^M$, from (2.19), and therefore

$$\bigcup_{x \in E} \text{Inv}(\Gamma_x^M) \subseteq \mathcal{C}^M. \quad (2.28)$$

We will now show that

$$\mathcal{C}^M \subseteq \bigcup_{x \in E} \text{Inv}(\Gamma_x^M), \quad (2.29)$$

by proving that any element of \mathcal{C}^M is included in the right-hand side of (2.29). Obviously this holds for \emptyset . We now verify that for any non-empty element $C \in \mathcal{C}^M$, $C \in \text{Inv}(\Gamma_x^M)$ for all $x \in C$. If C is a singleton, this is obvious because Γ_x^M is a connectivity opening (property 2, Theorem 1). Otherwise, C is subset of a grain of M , i.e. for any $x \in C$ we have $C \subseteq \Gamma_x(M)$, and $C \in \text{Inv}(\Gamma_x^M)$ for all $x \in C$. Thus every element of \mathcal{C}^M is contained in the union of the invariance domains of operators from the family $\{\Gamma_x^M \mid x \in E\}$. Therefore, (2.29) is true and, with (2.28), we have

$$\bigcup_{x \in E} \text{Inv}(\Gamma_x^M) = \mathcal{C}^M. \quad (2.30)$$

The connectivity opening of (2.20) can be used for both clustering or partitioning by generating the connectivity mask with an appropriate operator ψ from X . From (2.13) and (2.20), it is obvious that connectivity openings $\Gamma_x^\psi(X) = \Gamma_x^M(X)$, if $M = \psi(X)$ for any $x \in E$ and any $X \in \mathcal{P}(E)$, and that the resulting attribute filters will yield the same result. However, the connectivity classes are not the same. Consequently the connectivity class \mathcal{C}^M cannot be expressed explicitly as a superset or subset of the original \mathcal{C} .

2.4.2 Gray-Scale Mask-Based Attribute Filters

Attribute openings as mentioned earlier apply a trivial opening Γ_Λ on the output of a binary connectivity opening Γ_x . To associate these operators with the connected sets of a second-generation connectivity class we replace Γ_x with the corresponding connectivity opening [55], which in the case of mask-based connectivity is Γ_x^M . The increasingness of Γ_Λ and Γ_x^M makes it possible to extend Γ_M^Λ directly to gray-scale by the principle of *threshold superposition* [43]. Superimposing threshold sets requires their hierarchical nesting along the gray-scale. Given that a gray-scale image f can be decomposed to a set of binary images $T_h(f)$ resulting from thresholding f at all levels $h \in [0, N - 1]$, i.e.

$$T_h(f) = \{x \in E \mid f(x) \geq h\}, \quad (2.31)$$

the nesting of $T_h(f)$ within $T_k(f)$ is trivial for any $h \geq k$. In operator-based second-generation connectivity with ψ being any of the increasing structural operators described in Section 2.2.2, the nesting of $\psi(T_h(f)) \subseteq \psi(T_k(f))$ is obvious since the connectivity class \mathcal{C}^ψ applies to all threshold sets [10]. In mask-based connectivity however this is not the case; a different mask applies to every threshold set, and therefore a different connectivity class.

Let $m = \varphi(f)$ be a gray-scale connectivity mask where φ is an arbitrary operator, with $M_h = T_h(m)$ denoting each threshold set of m . Obviously $M_h \subseteq M_k$ for any $h \geq k$ (even if φ is non-increasing). At each level h the family of connectivity openings given in Proposition 1 yields a set of connected components according to \mathcal{C}^{M_h} . Because any connected component P_h^i of $T_h(f)$ must remain connected at lower grey levels (see [10] for details), we require that $\mathcal{C}^{M_h} \subseteq \mathcal{C}^{M_k}$ for any $h \geq k$. Since $M_h \subseteq M_k$ all we need to prove is the following proposition:

Proposition 2. *For any two mask-based connectivity classes \mathcal{C}^M and \mathcal{C}^L associated to connectivity masks $M \subseteq L \subseteq E$, the following property holds:*

$$\mathcal{C}^M \subseteq \mathcal{C}^L. \quad (2.32)$$

Proof. Comparing two mask-based connectivity classes as given in (2.19) is by looking at the nesting of the grains. For two connectivity masks such that $M \subseteq L$, the nesting implies that:

$$\forall C_i^M \exists j : C_i^M \subseteq C_j^L \quad (2.33)$$

in which C_i^M and C_j^L are connected components of M and L respectively. Therefore for any set $A \subseteq E$:

$$A \subseteq C_i^M \Rightarrow A \subseteq C_j^L. \quad (2.34)$$

Therefore, if $A \in \mathcal{C}^M$, this implies $A \in \mathcal{C}^L$, i.e., $\mathcal{C}^M \subseteq \mathcal{C}^L$.

Superimposing the outputs of the filtered threshold sets can be summarized to:

Definition 8. *For a mapping $f : E \rightarrow \mathbb{R}$, the gray-scale mask-based attribute opening $\gamma_m^\Lambda(f)$ is given by:*

$$(\gamma_m^\Lambda(f))(x) = \sup\{h \mid x \in \Gamma_\Lambda(\Gamma_x^{T_h(m)}(T_h(f)))\} \quad (2.35)$$

Thus, the mask-based attribute opening of a gray-scale image assigns each point of the original image the highest threshold at which it still belongs to a connected foreground component. Similarly, using a non-increasing criterion Λ , we can define the mask-based attribute thinnings.

2.5 Computing Second-Generation Attribute Filters

2.5.1 The Max-Tree Algorithm

The Max-Tree is a hierarchical image representation algorithm introduced by Salembier et al. [65] in the context of anti-extensive attribute filtering. The tree structure encodes the set-theoretical notion of connectivity and its gray-scale extension within the nesting properties

of the level components which are represented by nodes. It resembles to a certain extent the *Component Tree* by Jones [35] and its derivative *Gray-scale Component Tree* by Braganeto et al [10] where the nodes of the first correspond to binary level-sets while those of the second to gray-scale images. Connectivity at multiple scales is modelled by the *Connectivity Tree* of Tzafestas and Maragos [83] but can only handle binary images. Our work on second-generation connectivity representation and filtering is based on the Max-Tree primarily due to the algorithm's ability to handle non-increasing attributes on gray-scale images at rather low computational time.

The Max-Tree nodes correspond to connected components or sets of flat zones and there exists a unique mapping to *peak components*. A peak component P_h at level h is a connected component of the thresholded image $T_h(f)$ and a *regional maximum* M_h at level h is a level component no members of which have neighbors of intensity larger than h . The regional maxima in this case correspond to the leaves of the tree. Each tree node C_h^k (k is the node index) corresponding to a certain peak component contains only those pixels in P_h^k which have gray-level h . In addition each node except for the root, points towards its parent $C_{h'}^{k'}$ with $h' < h$. The root node is defined at the minimum level h_{\min} and represents the set of pixels belonging to the background.

The attributes of the connected components are computed during the construction phase of the tree and stored within the corresponding node structure. The attributes can be either increasing or non-increasing such as area/volume or shape descriptors such as moment invariants respectively. In both cases the peak component k at level h inherits the attribute data of all the peak components $P_{h'}^{k'}$ connected to C_h^k at levels $h' > h$. Thus, computing an attribute filter reduces to removing all nodes with attribute value smaller than a given threshold λ from the tree. Note that the node filtering is a separate stage from the computation of attributes and connected component analysis [65] and consumes only a short fraction of the total computation time.

2.5.2 The Dual-Input Max-Tree Algorithm

The Dual-Input Max-Tree algorithm presented in this section operates like the conventional Max-Tree [47] only it requires two input images; the original image X and the connectivity mask M according to Proposition 1. The tree is constructed in a recursive manner from data retrieved from a set of hierarchical first in - first out (FIFO) queues. The queues are allocated at initialization in the form of a static array called *HQueue* segmented to a number of entries equal to the number of gray levels in the connectivity mask. Data are accessed and stored in each queue entry by the flooding function (Fig. 2.7) which re-assigns priority pixels to the Max-Tree structure and stores new pixels retrieved from the neighborhood of the one under study, to the appropriate queue entries.

The Max-Tree structure consists of nodes corresponding to pixels of a given peak com-

```

flood(h, thisAttribute) {
  attr = InitializeAttribute() /* Initialize attr */
  if(thisAttribute) /* Accounts for child attributes */
    MergeAuxData(attr, thisAttribute)

  while (not HQueue-empty(h) /* First step: propagation */
  { p = HQueue-first(h) /* Retrieve priority pixel */
    STATUS[p] = NumberOfNodes[ORI[p]] /* STATUS = the node index */
    x = x_coord_of_p /* Retrieve x, y, z coordinates of p */
    y = y_coord_of_p
    z = z_coord_of_p

    if(ORI[p]!=h) { /* Detect intensity mismatch */
      NodeAtLevel[ORI[p]]=TRUE /* Same for both cases */
      idx = NodeOffsetAtLevel[ORI[p]] /* Get the parent node offset */
      if(Tree[idx]==NULL) /* Check if node exists, if not create */
        { Tree[idx] = malloc(...)
          Tree[idx]->Attribute = NewAuxData(x,y,z)
        } else
          AddAuxData(Tree[idx]->Attribute, x,y,z)
      if(ORI[p]>h){ /* Local partitioning */
        Tree[idx]->Parent = NodeOffsetAtLevel[h] /* Finalize Singleton node */
        Tree[idx]->Status = Finalized
        Tree[idx]->Level = ORI[p]
        NumberOfNodes[ORI[p]] += 1;
        NodeAtLevel[ORI[p]] = FALSE
        AddAuxData(attr,x,y,z) /* Update current node's attribute */
      } else
        AddAuxData(attr,x,y,z) /* Same pixel intensity in both images */
      for (every neighbor q of p) /* Process the neighbors */
        { if (STATUS[q] == "NotAnalyzed")
          { HQueue-add(P_ORI[q],q) /* Add in the queue */
            STATUS[q] = "InTheQueue"
            NodeAtLevel[P_ORI[q]] = TRUE /* Confirm node existence */
            if (P_ORI[q] > P_ORI[p]) /* Check for child nodes */
              { m = P_ORI[q]
                child_attribute = NULL
                do{ /* Recursive child flood */
                  m = flood(m,child_attribute)
                } while (m != h)
                MergeAuxData(attr,child_attribute)
              }
          }
        }
    }
  }

  NumberOfNodes = NumberOfNodes[h] + 1 /* Update the node index */
  m = h-1 /* 2nd step: defines father */
  while ((m >= 0) and (NodeAtLevel[m] = FALSE)) m = m-1;
  if (m >= 0){ /* Node parent is not the background */
    idx = NodeOffsetAtLevel[h] -1 /* Check if node exists, create if not */
    if(Tree[idx]==NULL) { ... } /* (as in the ORI[p]!=h case) */
    Tree[idx]->Parent = NodeOffsetAtLevel[m] /* Compute the parent node */
  } else { /* Node parent is the background */
    idx = NodeOffsetAtLevel[h] /* Check if node exists, create if not */
    if(Tree[idx]==NULL) { ... } /* (as in the ORI[p]=h case) */
    Tree[idx]->Parent = idx /* Compute the parent node */
  }
  MergeAuxData(Tree[idx]->Attribute, attr) /* Merge node attributes */
  Tree[idx]->Status = Finalized /* Finalize node */
  Tree[idx]->Level = h
  NodeAtLevel[h] = FALSE
  thisAttribute = Tree[idx]->Attribute /* Set 'thisAttribute' for parent node */
  return (m) }

```

Figure 2.7: The flooding function of the Dual Input Max-Tree algorithm adopted for attribute openings and thinnings. The parameters h and m are the current and child node gray levels while $attr$ is a attribute count at level h within the same connected component. The parameter $thisAttribute$ is used to pass child attributes to parent nodes.

ponent P_h^k at level h . Each node is characterized by its level h and index k and contains information about its parent node id, the node status and the attribute value (note that the tree structure is shaped by the histogram of the original image).

The two structures are managed with the aid of three arrays; the $STATUS[p]$, the $NumberOfNodes[h]$ and the $NodeAtLevel[h]$. $STATUS$ is an array of image size that keeps track of the pixel status. A pixel p can either be *NotAnalyzed*, *InTheQueue* or already assigned to node k at level h . In this case $STATUS[p]=k$. The $NumberOfNodes$ is an array that stores the number of nodes created until that moment at level h . Last, $NodeAtLevel$ is a boolean array that flags the presence of a node still being flooded at level h .

Initialization

During initialization, the status of all image pixels is set to *NotAnalyzed*. Similarly the $NumberOfNodes$ is set to zero while $NodeAtLevel$ is set to *FALSE* for each gray level. The histograms of both images are then computed to shape the $HQueue$ and Max-Tree structures accordingly. The first pixel with the lowest intensity h_{\min} in M , retrieved during the histogram computation, is placed in the corresponding queue while the three arrays are updated. This pixel defines the root node and is passed on to the main routine (flood) as the starting element.

The Flooding Function

The flooding routine is a recursive function involved in the construction phase of the Max-Tree. It is initiated by accessing the first root pixel from the queue at level h_{\min} and proceeds with flooding nodes along the different root paths that emerge during this process. The pseudo-code in Fig. 2.7 describes in detail the steps involved. ORI and P_ORI are two image-size arrays containing the pixel intensities in the original image and connectivity mask respectively.

Calling this function for a given level h , it first initializes an attribute variable `attr` which is updated for every pixel at level h in both ORI and P_ORI . An inspection on pixel availability for the given queue entry at level h proceeds by retrieving the first available pixel and continues with flooding, otherwise it skips flooding and finalizes the current node. If a pixel is available, its status is updated to the current node index for the level at $ORI[p]$ and its coordinates are computed. The process until this instance is identical to the conventional Max-Tree algorithm. The Dual Input Max-Tree upon retrieving a pixel inspects for an intensity mismatch between the ORI and P_ORI entries. If $ORI[p] < P_ORI[p]$ where p is the pixel under study, the connectivity mask involves local clustering while if $ORI[p] > P_ORI[p]$ it involves local partitioning.

The first case implies that p is a background pixel in the original image therefore it is regarded as connected to the current active node at level $ORI[p]$ through the connected

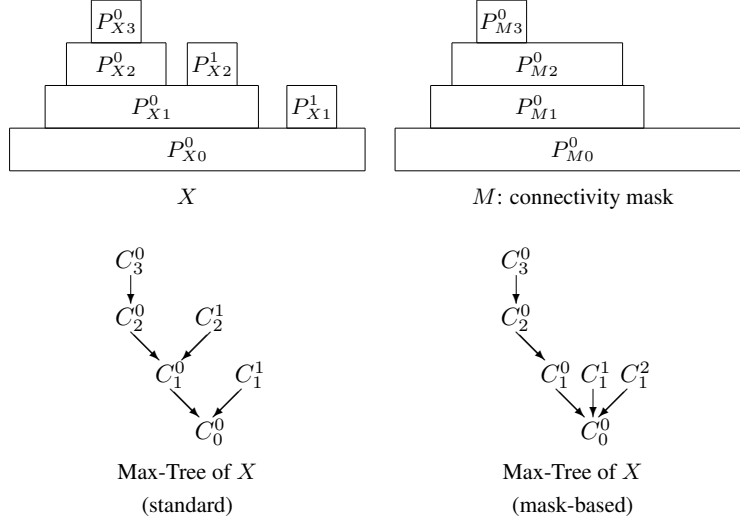


Figure 2.8: Dual Input Max-Tree: The attributes of C_2^0 and C_2^1 are merged to C_2^0 since all pixels at level $h = 2$ are clustered to a single peak component. Furthermore C_1^1 breaks up to a number of singleton nodes equal to the number of pixels in P_{M1}^1 .

component at level $P_ORI[p]$, i.e. it defines a peak component at level $ORI[p]$ to which p in the modified image is connected. $NodeAtLevel[ORI[p]]$ is set and a subroutine inspects for a node allocated for that level. If not found, a node is created and its attribute is initialized otherwise we simply update the existing node attribute.

In the second case where partitioning is involved we have $P_ORI[p] < ORI[p]$. Pixel p is therefore part of a discarded component according to definition 1, and consequently is treated as a singleton. Singletons define a node of unit attribute at level $ORI[p]$ hence upon detection the node must be finalized before retrieving the next priority pixel from the corresponding queue at level $P_ORI[p]$. This involves setting the node status to the node index at level $ORI[p]$ and detecting the parent node id. The attribute is initialized and upon completion $NodeAtLevel[ORI[p]]$ is set to *FALSE* indicating that this node is finalized. Note that since the singleton nodes are not flooded they do not return their attributes to the their parent node by default hence `attr` at the current level must be updated separately.

Both cases are demonstrated in Fig. 2.8 using a 1-D example. The first two diagrams illustrate the nesting of peak components in the original image X and the connectivity mask M . Note that M was chosen such that P_{X2}^0 and P_{X2}^1 from X are clustered to a single peak component while P_{X1}^1 vanishes. This results in replacing C_2^0 and C_2^1 with a single node in which the attributes of the two components are merged and splitting the node C_1^1 to a number

of singleton nodes equal to the number of pixels in P_{X1}^1 (illustrated in the last diagram).

If there is no mismatch between *ORI* and *P_ORI* we simply update `attr` and proceed with inspecting the neighborhood of p . The number of neighbors depending on the foreground connectivity is stored temporarily in a dynamically allocated array from which we retrieve them sequentially and inspect their status. If the status of a neighboring pixel q is set to *NotAnalyzed*, its placed in the appropriate queue entry at level $P_ORI[q]$ and its *STATUS* and *NodeAtLevel* attributes are updated accordingly. This process terminates there unless the pixel q is at a higher level from p . In that case flooding halts at level $P_ORI[p]$ and a recursive call to the same function initiates flooding at level $P_ORI[q]$. This is repeated until reaching the regional maximum along the given root path. Once a node is flooded, there are no more pixels in the queue for the given level therefore the algorithm proceeds with parent detection and node finalization. Note that as opposed to the conventional Max-Tree, the node attribute is merged with `attr` since the last one is updated by pixels at the same level or by singletons at higher levels while the node attribute is possibly updated by local clusters already flooded at higher levels in *P_ORI*. A parameter `thisattr` is also updated with the overall node attribute and returns to the calling function (usually the flood of the parent). The Max-Tree structure is completed when all nodes are finalized.

Masks by Operators With Non-Flat Structuring Elements

Connectivity masks generated by operators with non-flat structuring elements often introduce new gray-levels or remove existing ones. The Dual-Input Max-Tree algorithm structure is based on the histogram of the original image and nodes generated on gray-levels that are not present in *ORI* overlap with others creating a memory conflict. To counter this, on the initialization of the Max-Tree structure we allocate a total of twice the image size entries for Max-Tree nodes and segment the structure based on the sum of the maximum number of pixels in *ORI* and *P_ORI* for each level.

In the case where gray-levels are removed, a further routine is required to handle a possible intensity mismatch between h_{\min} in *ORI* and in *P_ORI*. If h_{\min} in *P_ORI* is smaller than that in *ORI* no action is required since all nodes will be finalized during the flooding procedure. If however the opposite is true, i.e. h_{\min} in *P_ORI* is higher than in *ORI*, then the flood function will stop when it reaches h_{\min} in *P_ORI* on all nodes at gray-levels below it will remain non-finalized. Furthermore, the structure will not have a root node. To counter this when reaching the only node at h_{\min} in *P_ORI*, we reduce by one the updated node counter since no other nodes can be found at this intensity. A post process flag which is set if this mismatch occurs during the computation of the image histograms, triggers an additional routine that follows the tree flooding. Starting from h_{\min} in *P_ORI* to h_{\min} in *ORI*, the only one node that can exist per level in this margin is detected, its attribute measure is updated, the parent node is set and it is finalized in the same way as in the flood function.

Attribute Management

Attributes are managed by the use of four different functions; *InitializeAttribute()*, *NewAuxData()*, *AddAuxData()* and *MergeAuxData()*. Our implementation demonstrates two types of attribute filtering, area/volume openings and elongation filtering based on moment invariants. To handle both we use a structure called *InertiaData* made of the area/volume count and four sums namely *SumX*, *SumY*, *SumZ* and *SumSquares*. For 3-D datasets the shape filter is described in section 2.2.3 and in [94].

InitializeAttribute() simply allocates a structure of size *InertiaData* and initializes all members to zero. *NewAuxData()* does similar but initializes area/volume to 1 and sets the four sums to the given coordinates. *AddAuxData()* updates the area count by 1 and adds the *x*, *y*, *z* coordinates to the corresponding sums. *SumSquares* is updated by the sum of the squared *x*, *y*, *z* coordinates. Lastly *MergeAuxData()* merges two structures of size *InertiaData*, the first corresponding to the parent data and the second to the child data and sums the individual members accordingly. With the aid of these four functions the algorithm allows a number of other attributes to be computed this way.

2.5.3 Filtering and Image Restitution

The construction phase of the Dual-Input Max-Tree algorithm returns the same type of tree structure with the conventional Max-Tree. Routines for attribute filtering therefore do not differ between the implementations.

Filtering the Max-Tree constitutes a separate stage and involves visiting all nodes of the tree maximally twice. The node attributes are compared against a threshold λ and if the criterion as in (2.9) is not met the parent pointers of children of C_h^k are updated to point at the oldest "surviving" ancestor of C_h^k . The comparison is repeated until the criterion is satisfied. This is described as the *Direct Rule* [65] and has no further effect on the descendants of the filtered node. In contrast to this the *Subtractive Rule* from [85, 86, 94], classified as a non-pruning strategy lowers in gray value all the descendants by the same amount as C_h^k itself. Other filtering rules are described by Salembier et al. [63, 65].

The node attributes are computed upon visiting each node from the attribute data stored in the node structure during the construction phase of the tree. This is realized by a routine implementing the $I/V^{5/3}$ term explained in section 2.2.3.

The output image *Out* is generated by visiting all pixels *p*, retrieving their node ids from *ORI* [*p*] and *STATUS*[*p*] and assigning the output gray level of that node to *Out*[*p*].

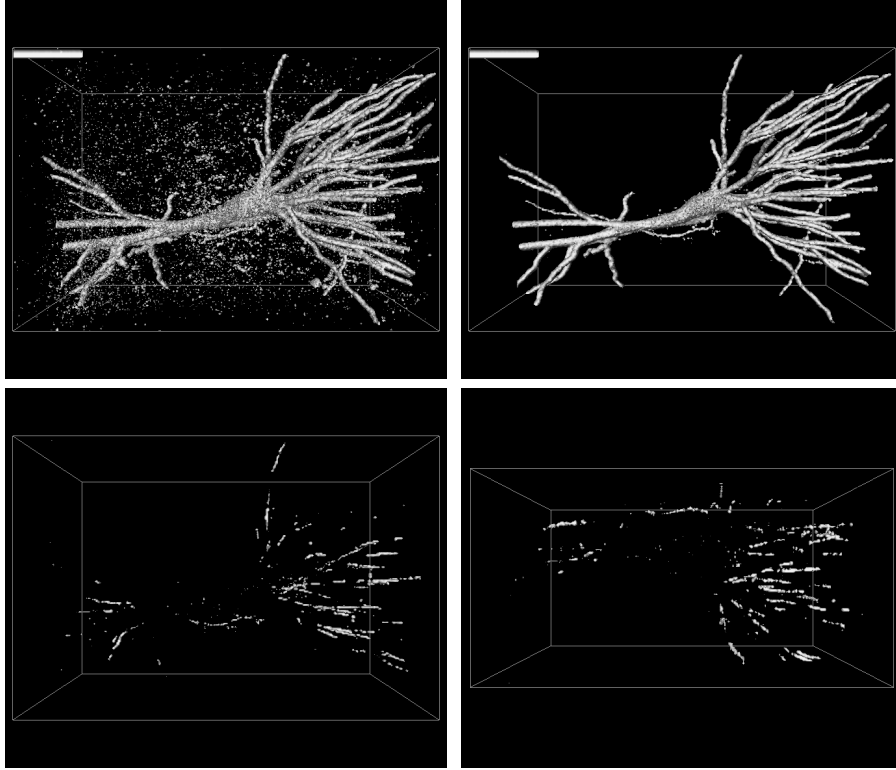


Figure 2.9: Top row: Isosurface projection of the neuron at level 1 (left); elongation filtering using Dual-Input Max-Tree algorithm (isolevel 1)(right). Bottom row: difference volume between the Dual-Input and conventional Max-Tree outputs - x - y and y - z views (isolevel 1).

2.6 Experiments and Discussion

The Dual-Input Max-Tree algorithm has been employed for area openings in [55] and in this paper an extension is presented to handle more complicated attributes such as the elongation measure discussed in Section 2.2.3 both in 2-D and 3-D. Furthermore, the new update supports connectivity masks generated by non-flat structural operators. The Dual-Input Max-Tree algorithm is derived from the conventional Max-Tree therefore it shares a number of characteristics concerning its performance which are discussed in depth in [47, 65]. If the same image is used twice in the Dual-Input Max-Tree algorithm, i.e. if $M = X$, it simply returns the Max-Tree of the original image.

In this section we first demonstrate the new features of the algorithm on a 3-D biomed-

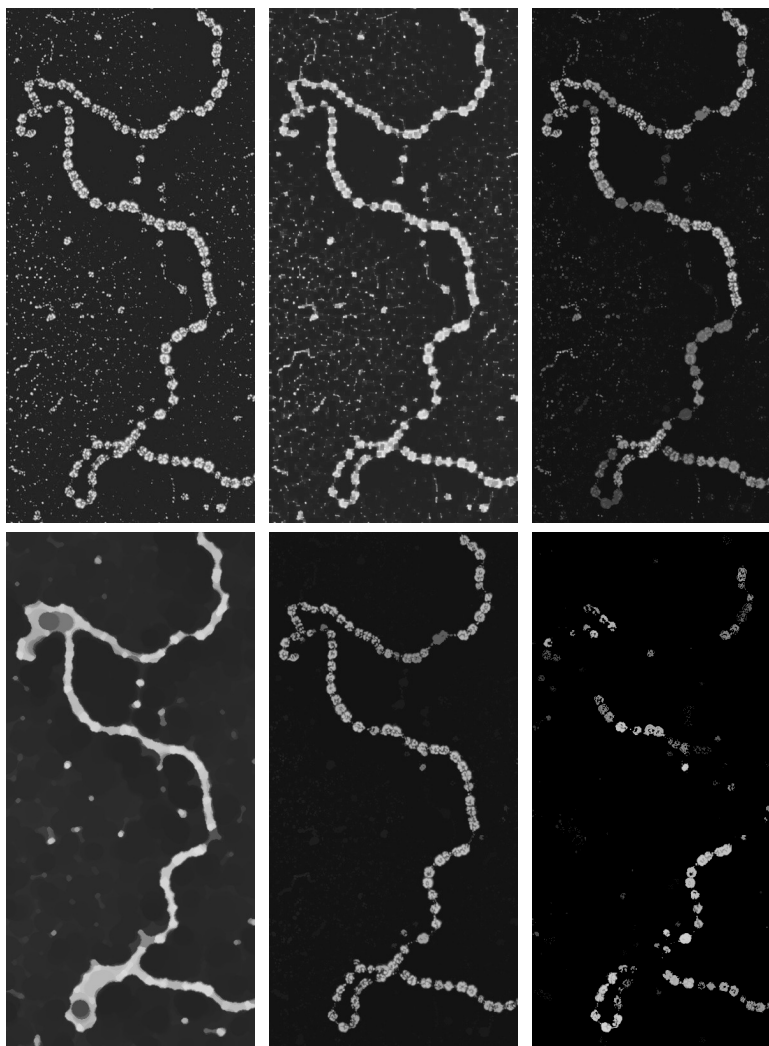


Figure 2.10: Top row: The original image; the mask by a closing with an SE of size 5×5 ; the filtered output with $\lambda = 3$. Bottom row: The mask by an ASF followed by an additional closing; the filtered output with $\lambda = 4$; and the difference image after contrast enhancement.

cal dataset and compare the result with that obtained using the same filter based on ordinary connectivity. The second part demonstrates the filtering improvements using various operators that were previously excluded due to constraints imposed by the earlier formulation of

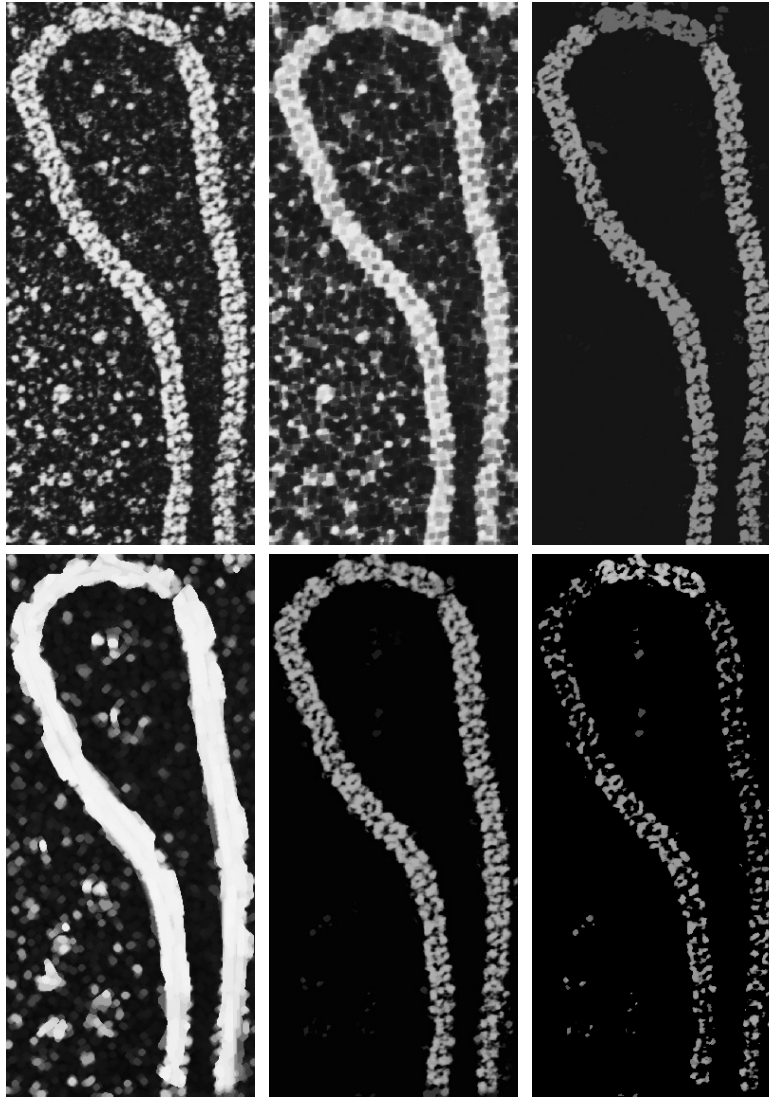


Figure 2.11: Top row: The original image; the mask by a closing with an SE of size 5×5 ; the filtered output with $\lambda = 6$. Bottom row: The mask described in the text; the filtered output with $\lambda = 6$; and the difference image after contrast enhancement.

the second-generation connectivity framework. The performance of the algorithm is evaluated by experiments on the 3-D dataset by measuring the CPU times for multiple runs and comparing it against the conventional Max-Tree. Dependencies of the algorithm are also discussed. All experiments were carried out on a 2.8 GHz Pentium IV CPU with 1 GB DDR memory. Our implementation was written in ANSI-C and the code is available upon request.

2.6.1 3-D Biomedical Datasets

This first experiment shows the applicability of the Dual-Input Max-Tree algorithm to the case of operator-based second-generation connectivity, in the case of a non-increasing attribute. Max-Trees have been employed for volume filtering and 3-D filament enhancement [94] of biomedical datasets. In this section we demonstrate a similar application with second-generation connected volumes. The algorithm uses the non-increasing 3-D shape filter based on the elongation measure of the filamentous structures (discussed in Section 2.2.3). All the illustrations are isosurface projections.

The dataset shown at the top left image of Fig. 2.9, is a $256 \times 342 \times 243$, 8-bit confocal microscopy volume of a pyramidal neuron. The noise density is relatively low but the filamentous structures (the dendrites in this case) are fragmented at low levels. Filtering using ordinary connectivity consequently removes noise together with a considerable fraction of the dendrites. If the volume is clustered however, nearby fragments are connected into a single entity with overall elongation greater than the threshold λ and hence are retained. The top right image shows the result of the shape filter based on clustering connectivity and using a cubic SE of size $5 \times 5 \times 5$. The two images of the bottom row show two different views of the difference volume between the filtering methods. Timings for this data sets were were 3.498 s for tree construction and 0.089 s for tree filtering using the conventional Max-Tree and 3.849 s and 0.061 s using the Dual-Input Max-Tree algorithm respectively.

2.6.2 Images of Proteins

In this subsection we demonstrate the use of AS filters and directional Minkowski additions with adaptive structuring elements for generating connectivity masks. All cases are compared with attribute thinnings based on clustering connectivity since the objective is to extract filamentous structures with disconnected members from the noisy background.

The first case is illustrated in Fig. 2.10. Generating a connectivity mask with either dilations or closings proves insufficient since both operators merge neighboring particles to the targeted object. The third image from the left - top row shows the result of elongation filtering using a mask (middle image) by a closing with a square SE of size 5×5 and $\lambda = 3$. We arrive at these settings since with any larger SE we cluster too much noise together with the protein chain and with any larger λ further parts of the chain are removed. If however after a closing we perform an opening with an SE of equal size all the thin bridges

introduced by the closing are removed. Applying a further closing by an SE of 13×13 clusters all desired members of the chain to large elongated fragments which are retained after filtering. The specific operator is both increasing and idempotent but neither extensive nor anti-extensive. The filtering improvement is evident in the second image of the bottom row and to highlight the difference between the two types of connectivity masks we compute the difference image and enhance it with contrast stretching. Concluding it can be seen that to avoid merging background noise, ordinary clustering-based masks are limited and therefore cannot capture the entire chain thus elongation filters with high attribute threshold flatten certain chain regions by removing higher-intensity components. Had we used a sequence of attribute filters, with alternating clustering-based and contraction-based connectivities, we would not obtain the same result. This is because the partitioning operator may remove small objects which are irretrievably lost to any consecutive clustering operation. We perform the clustering/partitioning sequence, and only after pixels have been grouped properly, decide what the attributes of these groups are. Any interim filtering could distort or even completely destroy a group we wish to retain.

The second case illustrated in Fig. 2.11 is handled with directional Minkowski additions. To generate the connectivity mask we first employ the Gabor wavelet based method of [25] to compute the predominant orientation along which to perform the addition. The kernel used is of a fixed size and we use 18 angle steps of 10° each. The convolution of the resulting wavelets with the input image is only used to compute the direction with the maximum filter response and no intensity modification takes place. Weak responses are ignored by thresholding the output while in cases where there is more than one response we handle each orientation separately. For each response above the threshold we perform a Minkowski addition using a line SE along the given orientation. Our implementation is based on the original algorithm by Soille et al. [77]. To counter thin line fragments of high elongation emerging at the background we compute an additional structural opening which for a square SE of size 3×3 yields a connectivity mask as seen in the first image of the bottom row in Fig. 2.11. The filtered output with an elongation threshold $\lambda = 6$ is given in the next image and the difference between this method and the equivalent result using the optimal clustering-based connectivity mask is shown in the last image after a log enhancement. The filtered outputs of both methods retain certain background elements which can be later removed with an area opening. The advantage of the idempotent, non-increasing and neither extensive nor anti-extensive operator presented as opposed to an ordinary closing is that it merges elements of the chain only and we can use large enough kernels to create a single object. By contrast, using a structural closing we face limitations similar to those mentioned in the previous example and consequently the protein chain remains fragmented. This results in severe flattening by the elongation filter which is illustrated in the difference image.

The images used in this section are courtesy of the Institute for Molecular Virology - University of Wisconsin, Madison, and can be obtained from the online Electron Micrograph

Library at <http://www.biochem.wisc.edu/inman/empics/index.html>.

2.6.3 Computational Complexity

The computational complexity of our implementation has a strong dependency on the image content and on the size of the structuring element used to create the connectivity mask. The content obviously affects the size of the tree, the number of recursions and therefore the time complexity.

The size of the SE affects the timing depending on the type of connectivity mask. For cases where M is generated by an extensive operator from X , the greater the size of the SE the lower the number of the connected components. Therefore building the Max-Tree should consume less time as the SE size increases. This however is not always the case with dynamically allocated attributes since the greater the number of mismatches between voxels in the two volumes the larger the number of searches for nodes at the parent level in *ORI*. With attributes represented by a scalar variable instead this is not true since no parent nodes need to be detected and allocated before finalizing the one being flooded.

If M is generated by an anti-extensive operator from X then the time overhead rises as the size of the SE increases since there are more singletons generated and hence more nodes that need to be processed. Dynamically allocated attributes contribute an additional delay as discussed above.

For cases where M results from a neither extensive nor anti-extensive operator from X or is independent of X , the performance of the algorithm depends strongly on the content of M . Means of evaluating it is by studying the frequency of occurrence of regions of M that appear as the result of an either extensive or anti-extensive operator with respect to X .

The filtering stage contributes a fixed time overhead which varies with the image/volume size. In all cases of M , filtering needs to access each pixel at most twice, therefore the search depth along different root-paths is compensated by reducing the number of remaining pixel visits.

2.7 Conclusions and Further Work

In this paper we presented an extension of the theory of second-generation connectivity. The connectivity opening introduced for this purpose is associated with connectivity masks rather than structural operators eliminating this way dependencies on their properties. This allows for images to be connected in any arbitrary fashion according to the connectivity mask and poses no restriction as to how the mask should be created. The main advantage is that any operator can be used to derive a second-generation connectivity class as opposed to the previous framework that was restricted to certain dilations, closings and openings only. Indeed, we could even use images of the same scene taken in different frequency bands (e.g.

optical/IR combinations), or using different imaging modalities (optical/range imaging, or registered CT/MRI pairs) to act as connectivity masks. When using a single filtering step, using a mask derived from the image by an arbitrary operator, the distinction between mask-based and operator-based connectivity may not seem very great. When trying to compute, e.g., granulometries using multiple filtering steps, the distinction is far more obvious. The difference is whether one considers mask generation and attribute filtering as a single operator (as in the operator-based case), or as two distinct steps (as in the mask-based case). If we use operators which do not meet the requirements imposed by the operator-based framework, but insist on interpreting mask-generation as part of the resulting filter, the question arises what the properties of such filters are. Whilst our theory allows the use of any operator for mask generation, it is not concerned with the properties of the resulting combined mask-generation/filtering operator. By studying these combinations, we may find operators, beyond the known examples, which actually yield an operator-based second-generation connectivity, which is neither clustering nor partitioning. This is the topic of future work.

This theoretical work is complemented by, and indeed validates, an efficient algorithm for attribute filtering using mask-based connectivity, referred to as the Dual-Input Max-Tree algorithm, which is demonstrated on both 2-D and 3-D datasets. The algorithm is an extension of the conventional Max-Tree [65]. The current version supports connectivity masks generated with both flat and non-flat structuring elements and provides the functionality for a wider range of attributes to be computed.

Potential applications of this work include filtering and segmentation of datasets characterized by thin elongated structures (like the neuron demonstrated in the previous section), connected component analysis and processing of second-generation connected sets and flexible attribute management depending on the image context. The relatively low computational requirement in 2-D examples makes it possible to use the presented algorithm also in real-time applications such as motion detection/analysis, tracking and decision making tasks.

Future work on this area involves deriving connected operators that can counter the oversegmentation effect in the case of partitioning [54, 91, 92], with extensions to gray-scale as well as algorithmic methods for their efficient computation. The issue of noise clustering is also being investigated and currently we are working on attribute-based clustering techniques that will allow the algorithm to cluster only objects of similar structural characteristics.

Chapter 3

Filament Enhancement by Non-Linear Volumetric Filtering using Clustering-Based Connectivity

Leave no stone unturned.

Euripides, Heraclidae, circa 428 B.C.

Abstract

Shape filters are a family of connected morphological operators that have been used for filament enhancement in biomedical imaging. They interact with connected image regions rather than individual pixels, which can either be removed or retained unmodified. This prevents edge distortion and noise amplification, a property particularly appreciated in filtering and segmentation. In this paper we investigate their performance using a generalized notion of connectivity that is referred to as “clustering-based connectivity”. We show that we can capture thin fragmented structures which are filtered out with existing techniques.

3.1 Introduction

BIOMEDICAL data sets often contain curvi-linear, dendritic or other filamentous structures of interest which are susceptible to acquisition noise. Enhancing these structures can be of particular importance to certain medical applications and many methods have been proposed [53]. Some common drawbacks among them is noise amplification and edge distortion while they can also be computationally expensive.

In mathematical morphology, a family of operators called *connected filters* has been developed which interact with regions characterized by some notion of connectivity. According to these filters, *connected regions* can either be removed or retained unmodified based on a pre-specified attribute (shape in this case) but new edges cannot emerge. This edge and therefore shape-preserving property makes connected filters competitive to existing morphological methods for filament enhancement such as the multi-scale approach in [94].

The objects targeted are thin, plate-like (Fig. 3.1) and elongated structures which are often fragmented at higher gray-levels according to the standard connectivity. We aim at

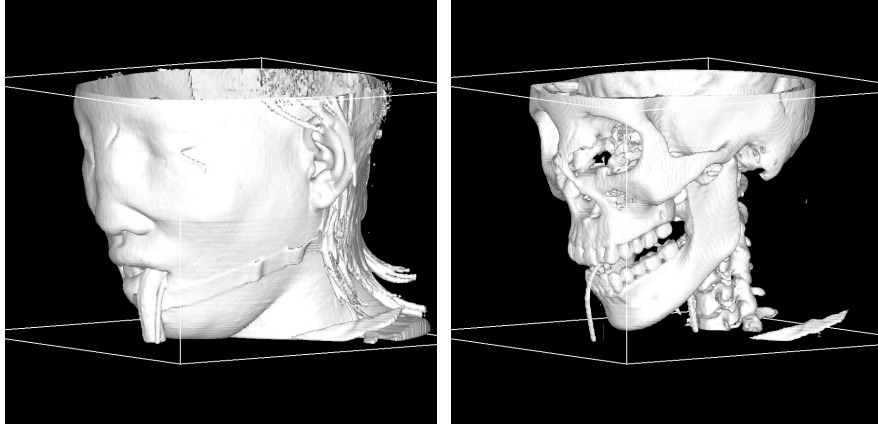


Figure 3.1: 3-D Shape filtering using 26 connectivity: The image on the left illustrates an isosurface projection of a human at isosurface level 208. Increasing the isosurface level to visualize the skull removes important details. The image on the right illustrates a shape filter enhancing the thin, plate-like structures comprising the skull and all the noise at an isosurface level 96.

countering this with a further improvement of the method presented in [94]. This is by using a more general notion of connectivity termed *clustering-based connectivity* [68, 69] which models object clusters as individual connected regions. We demonstrate our findings and compare them to the existing method using three different 3-D data sets. In each case we study the parameters which maximize the filter’s performance in association with the underlying clustering-based connectivity. Following this section there is a short reference to the concept of connectivity and connectivity openings complemented by the notion of clustering-based connectivity. In Section 3.3 the shape filters and their extensions to gray-scale are presented while in Section 3.4 we discuss their applications to 3-D medical data sets. The work is summarized with some conclusions in Section 3.5.

3.2 Theory

3.2.1 Connectivity Classes and Openings

The set-theoretic notion of connectivity in discrete spaces such as \mathbb{Z}^2 describes how groupings are realized in digital images. Connectivity in mathematical morphology is given by *connectivity classes*, a construct defined as:

Definition 9. Let E be an arbitrary (non-empty) set. A family $\mathcal{C} \subseteq \mathcal{P}(E)$ is called a connectivity class if it satisfies:

1. $\emptyset \in \mathcal{C}$ and for all $x \in E$, $\{x\} \in \mathcal{C}$,
2. for any $\{A_i\} \subseteq \mathcal{C}$ for which $\bigcap A_i \neq \emptyset \Rightarrow \bigcup A_i \in \mathcal{C}$

Members of \mathcal{C} are called *connected sets* [68, 69] and Definition 9 means that both the empty set and singleton sets are connected, and any union of connected sets which have a non-empty intersection is also connected.

Addressing objects in binary images is often more practical using *connected components* or *grains* which are connected parts of an object of maximal extend, i.e. they are connected and not smaller than any other connected part of the same object. Writing this explicitly, we say that C is a connected component of a binary image X if there is no set $C' \supset C$ such that $C' \subseteq X$ and $C' \in \mathcal{C}$.

Connected components are groupings of connected sets containing a certain point $x \in E$ in their intersection. The operator Γ_x to access them is called a *connectivity opening* marked by x and is given by:

$$\Gamma_x(X) = \bigcup \{A_i \in \mathcal{C} \mid x \in A_i \text{ and } A_i \subseteq X\}. \quad (3.1)$$

Furthermore, $\forall x \notin X$, $\Gamma_x(X) = \emptyset$. Connectivity openings are characterized by three properties; they are *anti-extensive*, *increasing* and *idempotent* operators. For a given set X each property implies the following:

1. Anti-extensiveness: $\Gamma_x(X) \subseteq X$,
2. Increasingness: if $X \subseteq Y \Rightarrow \Gamma_x(X) \subseteq \Gamma_x(Y)$,
3. Idempotence : $\Gamma_x(\Gamma_x(X)) = \Gamma_x(X)$.

The operator Γ_x is explicitly related to a connectivity class \mathcal{C} if satisfying the set of conditions given by Serra [68] (also in [62]) in the following theorem:

Theorem 2. *The datum of a connectivity class \mathcal{C} on $\mathcal{P}(E)$ is equivalent to the family $\{\Gamma_x \mid x \in E\}$ of openings on x such that:*

1. every Γ_x is an algebraic opening,
2. for all $x \in E$, we have $\Gamma_x(X) = \{x\}$,
3. for all $X \subseteq E$, $x, y \in E$, $\Gamma_x(X)$ and $\Gamma_y(X)$ are equal or disjoint,
4. for all $X \subseteq E$, and all $x \in E$, we have $x \notin X \Rightarrow \Gamma_x(X) = \emptyset$.

Connectivity openings characterize uniquely the connectivity class they are associated with and there is a one-to-one correspondence between the two.

3.2.2 Clustering-Based Connectivity

Connected components of X according to \mathcal{C} are separated by elements of the background. If however the distance separating them is smaller than the size of a given structuring element (SE), it is possible to define a *cluster* [7, 62, 69] in a *child* connectivity class \mathcal{C}^ψ , where ψ denotes a structural operator referred to as *clustering*. Following is a list summarizing the properties required to define a clustering:

1. ψ is increasing and extensive.
2. $\psi(\mathcal{C}) \subseteq \mathcal{C}$.
3. For a family $\{X_i\}$ in $\mathcal{P}(E)$ such that $\psi(X_i) \in \mathcal{C}, \forall i$, and $\bigcap_i X_i \neq \emptyset \Rightarrow \psi(\bigcup X_i) \in \mathcal{C}$.
4. ψ does not create connected components; i.e., if $\forall x \in \mathcal{C}, C = \Gamma_x(\psi(X)) \Rightarrow X \cap C \neq \emptyset$.
5. ψ treats the clusters of X independently; i.e., if $\forall x \in \mathcal{C}, C = \Gamma_x(\psi(X)) \Rightarrow \psi(X \cap C) = C$.

More details on each item are given in [7]. Typically, ψ is either a dilation or a closing and generates a mask image, called the *connectivity mask* by expanding X .

Definition 10. Let \mathcal{C} be a connectivity class in $\mathcal{P}(E)$ and ψ be an increasing and extensive operator on $\mathcal{P}(E)$. Then

$$\mathcal{C}^\psi = \{X \in \mathcal{P}(E) \mid \psi(X) \in \mathcal{C}\} \quad (3.2)$$

is a clustering-based connectivity class for which $\mathcal{C} \subseteq \mathcal{C}^\psi$.

If, for ψ the above five properties hold, and furthermore, $\psi(\emptyset) = \emptyset$ and

$$\psi(X \cap \Gamma_x(\psi(X))) = \Gamma_x(\psi(X)), \quad (3.3)$$

we have a *strong clustering* [7].

Definition 11. Let $\{\Gamma_x \mid x \in E\}$ be the connectivity openings associated with \mathcal{C} . If ψ is a strong clustering on $\mathcal{P}(E)$, the family of connectivity openings $\{\Gamma_x^\psi \mid x \in E\}$ associated to \mathcal{C}^ψ are given by

$$\Gamma_x^\psi(X) = \begin{cases} \Gamma_x(\psi(X)) \cap X, & \text{if } x \in X \\ \emptyset, & \text{otherwise} \end{cases} \quad (3.4a)$$

$$(3.4b)$$

In the following, every time we use the term clustering we mean a strong clustering.

3.3 Shape Filters

Filtering a binary image based on the attributes of its connected components requires a criterion T commonly given by:

$$T(C) = (Attr(C) \geq \lambda) \quad (3.5)$$

where $Attr$ is some attribute value of a connected component C and λ a pre-selected threshold. Components that satisfy (3.5) are retained while the rest are removed. Binary attribute filters in the anti-extensive case can be categorized to attribute openings or thinnings depending on whether the attribute criterion is increasing or not. The case that $Attr(C)$ is non-increasing implies that for any two nested components C_1 and C_2 ,

$$C_1 \subseteq C_2 \Rightarrow Attr(C_1) \leq Attr(C_2), \quad (3.6)$$

i.e. their attributes need not be ordered in the same way. Comparing the attribute value of a connected component against λ is by means of a trivial thinning Φ_T on the output of the connectivity opening of (3.1). The trivial thinning is an anti-extensive, idempotent and non-increasing operator defined as $\Phi_T : \mathcal{C} \rightarrow \mathcal{C}$ which for a connected component $C \in \mathcal{C}$ yields C if $T(C)$ is true, and \emptyset otherwise. Furthermore, $\Phi_T(\emptyset) = \emptyset$. For a binary image X , the attribute thinning is given by:

$$\Phi^T(X) = \bigcup_{x \in E} \Phi_T(\Gamma_x(X)). \quad (3.7)$$

Attribute thinnings sensitive to structures of a given shape are called *shape filters*. The filamentous structures that we investigate, are thin elongated structures that are characterized by a high trace of the moment of inertia tensor $I(C)$ compared to their volume $V(C)$. For 3-D data sets, $I(C)$ has a minimum for a sphere and increases rapidly as the object becomes more elongated [94]. It is defined as:

$$I(C) = \frac{V(C)}{4} + \sum_{\mathbf{x} \in \mathbf{C}} (\mathbf{x} - \bar{\mathbf{x}})^2 \quad (3.8)$$

and scales with size to the fifth power whereas the volume scales with the third power of the size. Therefore the ratio

$$Attr(C) = \frac{I(C)}{V^{5/3}(C)} \quad (3.9)$$

is a purely shape dependent attribute which together with (3.7) defines a filter sensitive to elongated shapes.

Connected filters in general rely on some notion of connectivity. In the case of (3.7) the term $\Gamma_x(X)$ relates the filter to the connectivity class \mathcal{C} and the connected components it returns are unique. Extending connected filters to sets characterized by second-generation

connectivity is by replacing the connectivity opening with the associated operator. For clustering-based connectivity this is Γ_x^ψ .

The cases in which the attribute criterion of a filter is increasing, like the volume of a 3-D connected component $V(C)$, extend to gray-scale trivially [55, 57] based on the principle of threshold superposition [43]. For the non-increasing, translation and shift invariant shape descriptor of (3.9), gray-scale attribute filters based on either type of connectivity can be computed efficiently using the *subtractive filtering rule* [86]. This is a non-pruning, tree-based filtering strategy in which if a tree node (corresponding to a connected component of the thresholded image at level h) is reduced in gray-scale, its descendants are lower by the same amount. It is realized on a tree structure for second-generation connectivity representation termed the *Dual-Input Max-Tree* algorithm that is based on [65] and extended details can be found in [55, 57]. The experiments that follow are based on this arrangement.

3.4 Experiments

In this section we experiment with the 3-D shape filter discussed in Section 3.3, using clustering-based connectivity. In this first approach to non-linear volumetric filtering using this specific type of second-generation connectivity, the objective is to enhance and extract filamentous details from a number of noisy biomedical data sets. The present study investigates the factors that affect the performance of the proposed filter. We identify five critical parameters namely: (i) the neighborhood of each volume element in 3-D, (ii) the size of the structuring element to be used, (iii) the type of clustering operator ψ , i.e. a dilation or a closing, (iv) the way the attributes are calculated (on X or $\psi(X)$) and (v) the attribute threshold used with the filter.

The first data set is an isosurface projection of an 8 bit, $256 \times 256 \times 256$ rotational b-plane CT-angiogram (CTA) of the arteries of the right half of a human head (Fig. 3.2). A contrast agent was used and an aneurysm is present. The volume contains a dense cloud of low intensity noise centered within the structures of interest. To generate the connectivity mask we consider the first three parameters listed earlier. For volume sets it is common to use a 26 neighborhood since a 6 neighborhood often results in "loosely" connected components. Masks generated by a dilation expand the original set creating a number of structures of previously disconnected elements. In noisy backgrounds, this can result in grouping the noise elements to high attribute structures and create connections with the structures of interest. Using structural closings instead, the unwanted connections between small objects tend to break apart while structures merged by wide bridging regions are maintained. This is illustrated at the middle row of Fig. 3.2 where the image on the left shows the response of an elongation filter with $\lambda=3$ using a mask based on a dilation with a cubic SE of size $3 \times 3 \times 3$. The image on the right is the response of the same filter using a mask by a structural closing instead. It is evident that a dilation even with a relatively small SE merges most

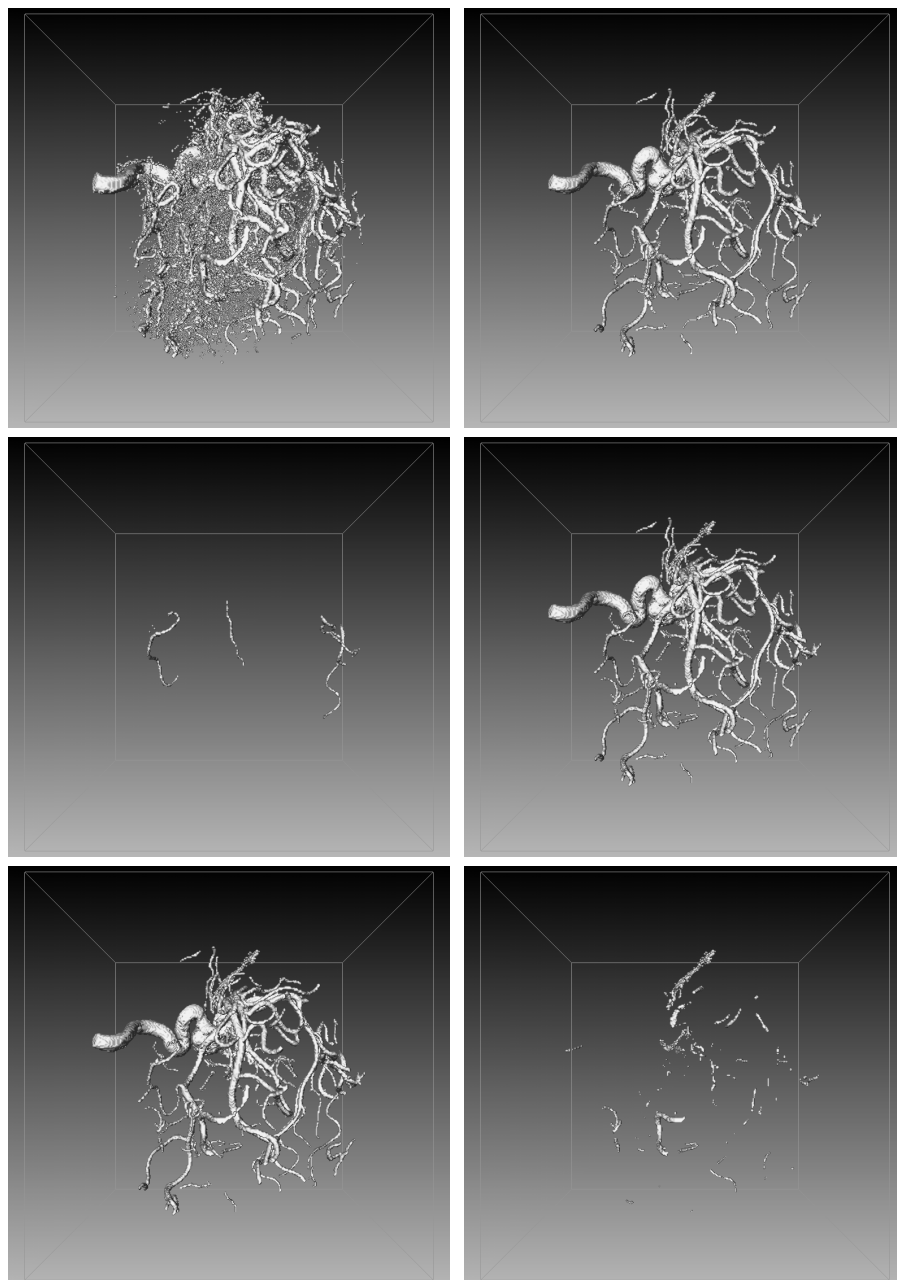


Figure 3.2: Isosurface projections of a CTA scan containing an aneurysm and the output of the elongation filter based on standard connectivity (both at isolevel 0). The middle row shows the filtered outputs using a mask based on a dilation and a closing respectively. The bottom row shows the difference volumes between the filter outputs using clustering-based connectivity based on a closing vs. a dilation and based on a closing vs. the standard connectivity. Most vessel-like structures are preserved using a closing-based connectivity.

of the noise together with the blood vessels creating a structure with large overall volume and small elongation. Filtering removes all but certain regions disconnected from the clustered volume. The results can be compared with the filter response using standard, 26-connectivity - top right image. The bottom row shows the difference volumes between the filter responses. In the left image we compare the responses using a closing and a dilation. It can be seen that most of the structure of interested is lost. The right image shows the difference in the response using a closing-based clustering connectivity and the standard connectivity. We see a number of elongated structures missed by the filter using standard connectivity. With the closing-based connectivity, these vessel fragments are merged with the overall structure and hence they are retained.

The second data set shown at the top left image of Fig. 3.3, is a $256 \times 342 \times 243$, 8-bit confocal microscopy volume of a pyramidal neuron. The noise density here is not as high as the previous data set, but the filamentous structures (the dendrites in this case) are fragmented at low levels. Filtering using standard connectivity removes noise together with a considerable fraction of the dendrites. If the volume is clustered however, nearby fragments are connected into a single entity with overall elongation greater than the threshold λ and hence are retained. The top right image shows the result of an elongation filter with $\lambda=2$ using the standard connectivity at a 26 neighborhood.

Creating a mask with a structural closing is often not sufficient to counter the issue of noise clustering. Noise can be clustered in arbitrary arrangements and along arbitrary orientations. Two examples are illustrated at the first two images of Fig. 3.4 where both clustered arrangements have a similar elongation measure (attributes computed on the clustered sets are referred to as *C-attributes*). If the elongation measure is computed based on the expanded sets as illustrated at the corresponding connectivity masks at the last two images, the attributes of the two clustered arrangements are separated by a larger margin that distinguishes easier compact from elongated clusters. Attributes computed on the expanded sets of the mask are referred to as *M-attributes*.

The two images of the middle row of Fig. 3.3 illustrate the filter response with a connectivity mask generated by a structural closing with a cubic SE of size $5 \times 5 \times 5$ and corresponding C- and M-attributes respectively. The difference volumes computed between the responses with C-attributes, and M-attributes vs. 26-connected filtering, respectively, are shown at the bottom row. It can be seen that together with a considerable fraction of the dendrites claimed by the filter based on clustering connectivity, computing M-attributes outperforms the output based C-attributes which fails to deal with clustered noise effectively. The top first four images are isosurface projections at level 1 and the last two at level 0.

The last data set is a $256 \times 256 \times 124$, 8-bit, phase contrast magnetic resonance angiogram (MRA) of a human head. In this experiment we target the blood vessels and experiment with the size of the SE to be used along ψ in generating the connectivity mask. The top left image of Fig. 3.5 shows the input volume at isolevel 50 (details start to appear only after this

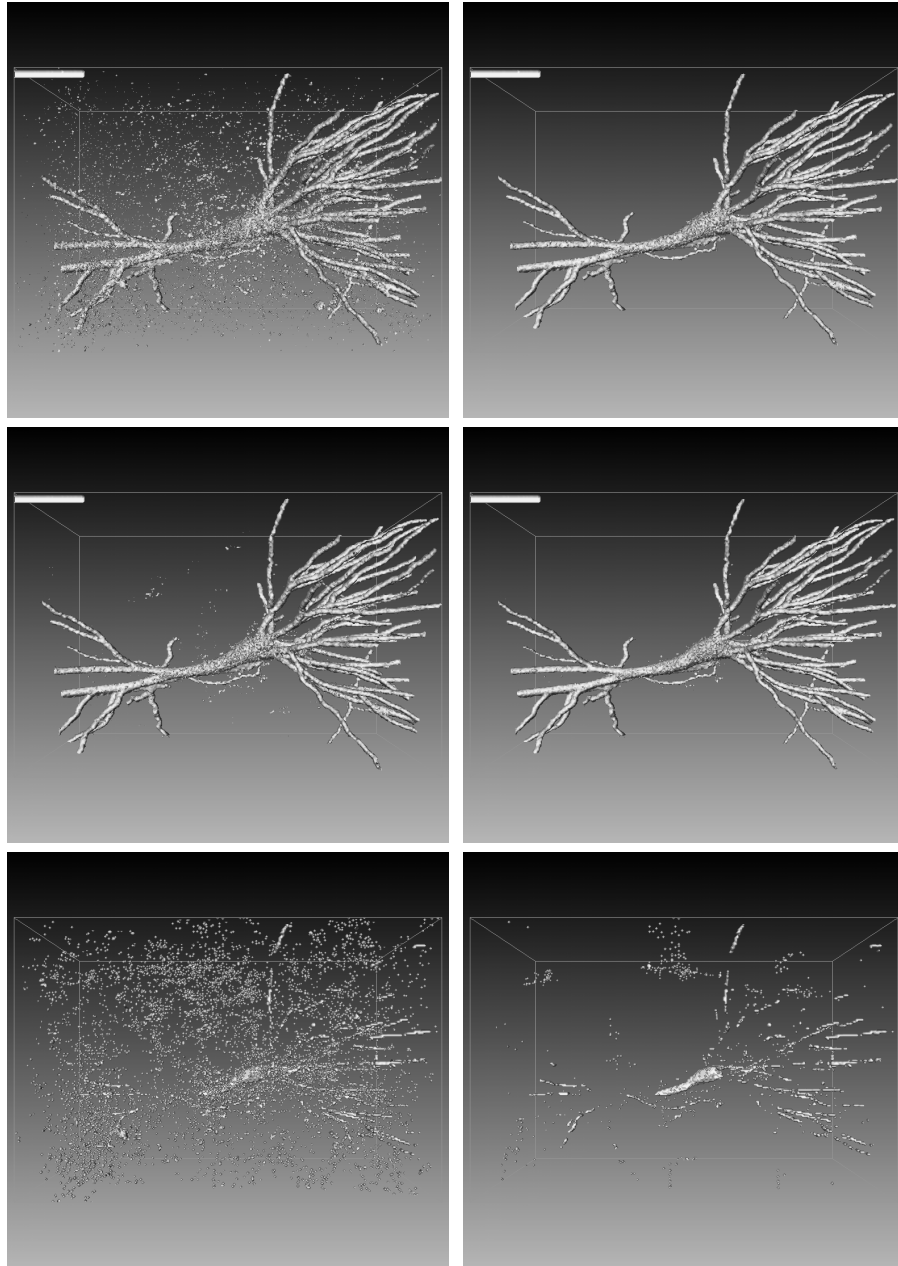


Figure 3.3: Isosurface projections of the neuron and the output of the elongation filter based on the standard connectivity, both at isolevel 1. The middle row illustrates the filter performance by computing the structure attributes based on the clustered volume and based on the expanded volume which constitutes the mask. The bottom row shows the difference volumes between the C-attributes vs. 26-connected filtering, and between the M-attributes vs. 26-connected filtering.



Figure 3.4: The elongation measures of the clustered sets X and Y (first and second image from the left) are similar if the we compute the C-attributes. The M-attributes instead are computed on $\psi(X)$ and $\psi(Y)$ (third and fourth image from the left respectively) and obviously the elongation of $\psi(X)$ is smaller compared to that of $\psi(Y)$.

threshold). The top right image and the two at the middle row (starting from the left) show the responses of an elongation filter with $\lambda = 2$ using standard connectivity, and clustering connectivity based on masks by a $3 \times 3 \times 3$ and $5 \times 5 \times 5$ cubic SE respectively (at isolevel 5). The filter uses M-attributes and from the difference volumes between the responses of the filter using clustering connectivity with 3^3 -based mask vs. standard connectivity and with 5^3 -based mask vs. standard connectivity, it can be seen that both deal relatively well with clustered noise (isolevel 1) and they both capture vessel fragments but at a varying detail. To examine their in-between differences we also compute the difference volume between the output with 3^3 -based mask vs. 5^3 -based mask and the reverse (Fig. 3.6). The left image illustrates that with an increasing size of SE, the overall signal intensity in the vessels is reduced, though there is no distortion. On the other hand as the size of the SE increases the number of fragments captured increases as well, as shown in the righthand image. This also contributes to some additional clustered noise. In general the size of the SE can only be determined by the amount of detail required and a quantitative evaluation is only possible given the ground truth.

3.5 Discussion

In this paper we compared the performance of connected filters for filament enhancement, based on classical connectivity and clustering-based connectivity. From the difference volumes produced in the previous section it can be seen that the 3-D shape filter, sensitive to elongated structures, captures filamentous details in greater accuracy when dependent upon an underlying clustering-based connectivity. This is because fragments of the filamentous structures are clustered with their original body, contributing to an overall elongation attribute greater than their own if treated separately.

The parameters influencing the performance of the filter have also been studied and we demonstrated how each one affects the filter response and in what way. A comparison with different elongation thresholds has not been carried out since it is obvious that as the value of λ increases the more elements will be filtered out. This can be useful for capturing highly elongated structures. In the case of blood vessels the handling of each vessel separately

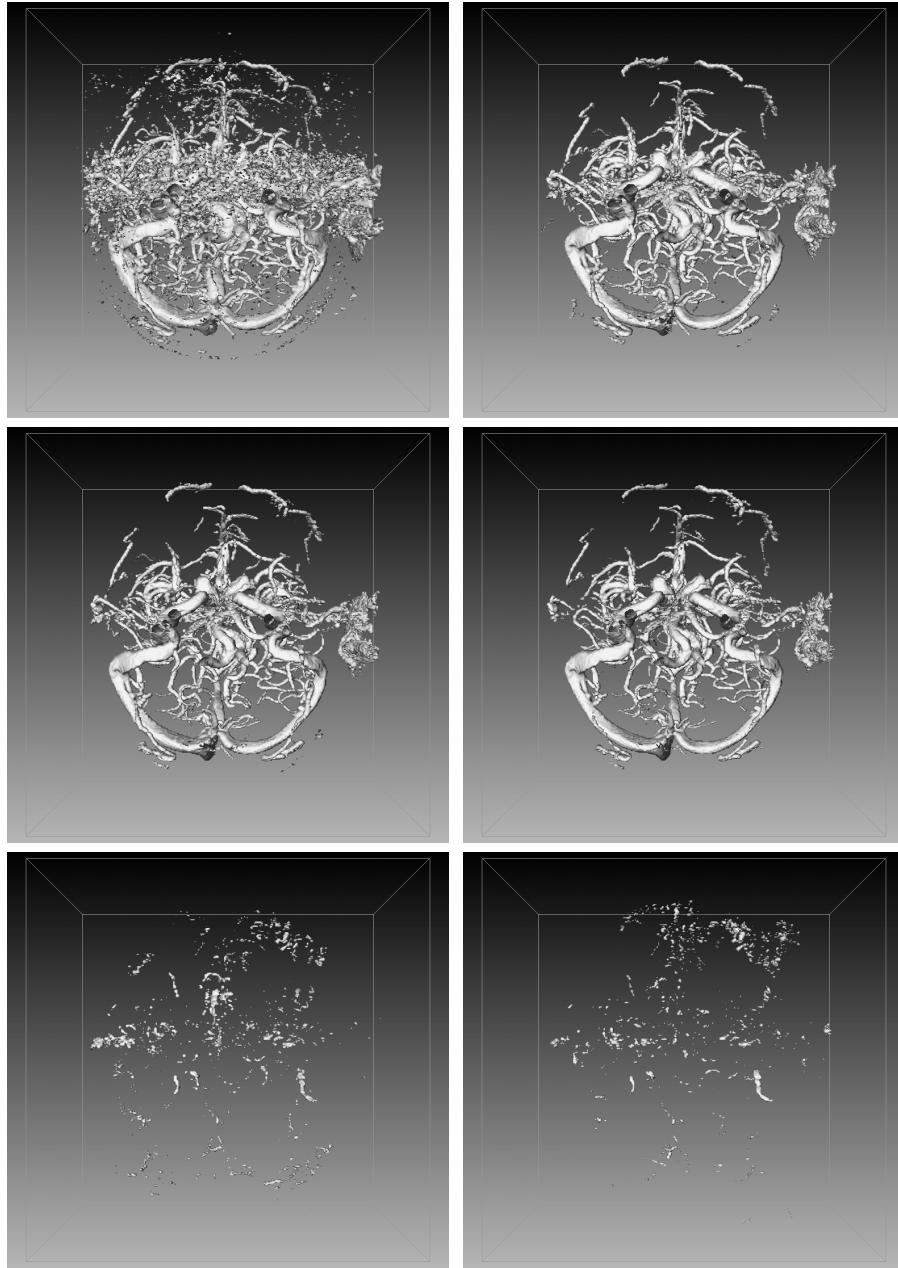


Figure 3.5: Isosurface projection of the MRA at isolevel 50 and the output of the elongation filter based on the standard connectivity at isolevel 5. The middle row illustrates the filter outputs using a clustering-based connectivity with masks generated by a structural closing with a cubic SE of size $3 \times 3 \times 3$ and $5 \times 5 \times 5$ respectively. The bottom row shows the difference volumes between the two filter outputs compared against the volume generated by the filter based on standard connectivity.

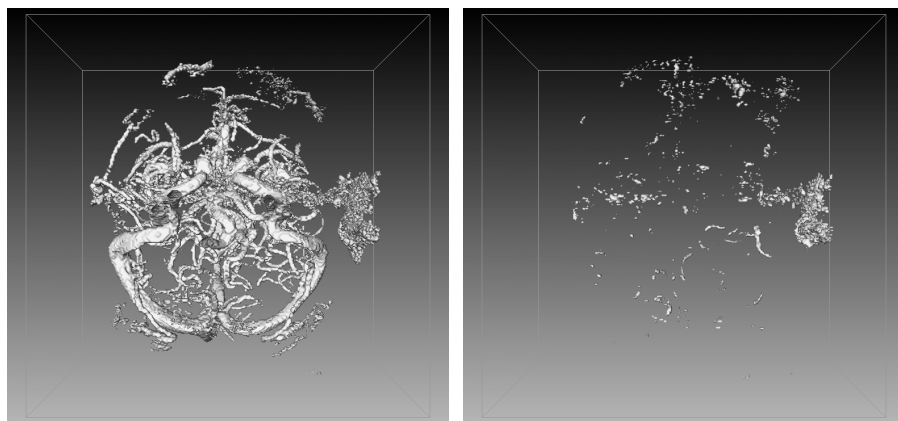


Figure 3.6: The difference volumes between a filter based on the 3^3 -based mask vs. the 5^3 -based mask, and the reverse, at isolevel 1.

involves a different type of second-generation connectivity called *contraction-based connectivity* which is not studied here.

A drawback of filters relying on a clustering-based connectivity is that of noise clustering. We minimize this effect by considering the structure attributes based on the connectivity mask instead of the clustered volume. We are currently working on further improvements by creating connectivity masks with adaptive structuring elements sensitive only to the direction of elongation.

Chapter 4

A Parallel Implementation of the Dual-Input Max-Tree Algorithm for Attribute Filtering

Let thy speech be short, comprehending much in a few words.

Aprocrypha

Abstract

This paper presents a concurrent implementation of a previously developed Dual-Input Max-Tree algorithm that implements anti-extensive attribute filters based on second-generation connectivity. The parallelization strategy has been recently introduced for ordinary Max-Trees and involves the concurrent generation and filtering of several Max-Trees, one for each thread, that correspond to different segments of the input image. The algorithm uses a Union-Find type of labelling which allows for efficient merging of the trees. Tests on several 3-D datasets using multi-core computers showed a speed-up of 4.14 to 4.21 on 4 threads running on the same number of cores. Maximum performance of 5.12 to 5.99 was achieved between 32 and 64 threads on 4 cores.

4.1 Introduction

ATTRIBUTE filters [11, 65] are a class of shape preserving operators. Their key property is that they operate on image regions rather than individual pixels. This allows image operations without distorting objects, i.e., they either remove or preserve objects intact, based on some pre-specified property. Attribute filters can be efficiently implemented using the Max-Tree algorithm [65], or similar tree structures [35, 89]

Image regions in mathematical morphology are characterized by some notion of connectivity, most commonly 4- and 8-connectivity. This yields an association between connectivity and connected operators which is extensively discussed in [9, 62, 69]. These papers also provide extensions to these basic connectivities known as *second-generation connectivity*. A general framework and algorithm is presented in [57]. The algorithm referred to as the *Dual-Input Max-Tree* supports the mask-based connectivity scheme, for which we give a concurrent implementation in this paper. It is based on the parallel Max-Tree algorithm in [93], which builds individual Max-Trees for image regions concurrently, and merges these trees efficiently.

4.2 Attribute filters

Attribute filters are based on *connectivity openings*. In essence, a connectivity opening $\Gamma_x(X)$ yields the connected component containing the point $x \in X$ and \emptyset otherwise. A connectivity opening is characterized by the following properties; for any two sets X, Y it is *anti-extensive* i.e. $\Gamma_x(X) \subseteq X$, *increasing* i.e. if $X \subseteq Y \Rightarrow \Gamma_x(X) \subseteq \Gamma_x(Y)$, and *idempotent* i.e. $\Gamma_x(\Gamma_x(X)) = \Gamma_x(X)$. Furthermore, for all $X \subseteq E, x, y \in E, \Gamma_x(X)$ and $\Gamma_y(X)$ are equal or disjoint.

A general approach in deriving second-generation connectivity openings using arbitrary image operators is given in [57]. A mask-based connectivity opening is defined as:

$$\Gamma_x^M(X) = \begin{cases} \Gamma_x(M) \cap X & \text{if } x \in X \cap M, \\ \{x\} & \text{if } x \in X \setminus M, \\ \emptyset & \text{otherwise.} \end{cases} \quad (4.1a)$$

$$\{x\} \quad \text{if } x \in X \setminus M, \quad (4.1b)$$

$$\emptyset \quad \text{otherwise.} \quad (4.1c)$$

where M is an arbitrary, binary mask image.

We can define a number of other connected filters based on a connectivity opening that work by imposing constraints on the connected components it returns. In the case of attribute openings such constraints are commonly expressed in the form of binary criteria which decide to accept or to reject components based on some attribute measure.

Attribute criteria Λ are put in place by means of a trivial opening Γ_Λ . The latter yields C if $\Lambda(C)$ is true, and \emptyset otherwise. Furthermore, $\Gamma_\Lambda(\emptyset) = \emptyset$. Attribute criteria are typically expressed as:

$$\Lambda(C) = Attr(C) \geq \lambda \quad (4.2)$$

with $Attr(C)$ some real-value attribute of C , and λ an attribute threshold.

Definition 12. The binary attribute opening Γ^Λ of a set X with an increasing criterion Λ is given by:

$$\Gamma^\Lambda(X) = \bigcup_{x \in X} \Gamma_\Lambda(\Gamma_x(X)) \quad (4.3)$$

Many examples are given in [11,65]. Note that if Λ is non-increasing we have an *attribute thinning* Φ^Λ [11] instead. An example is the scale-invariant non-compactness criterion of the form of (4.2), in which

$$Attr(C) = I(C)/V^{5/3}(C), \text{ where } I(C) = \frac{V(C)}{4} + \sum_{\mathbf{x} \in C} (\mathbf{x} - \bar{\mathbf{x}})^2 \quad (4.4)$$

with I the trace of the moment of inertia tensor in 3-D and $V(C)$ the volume of a component C [94]. Attribute filters can be operated on sets characterized by second-generation connectivity by replacing Γ_x with Γ_x^M instead. The proof of this and a more detailed analysis

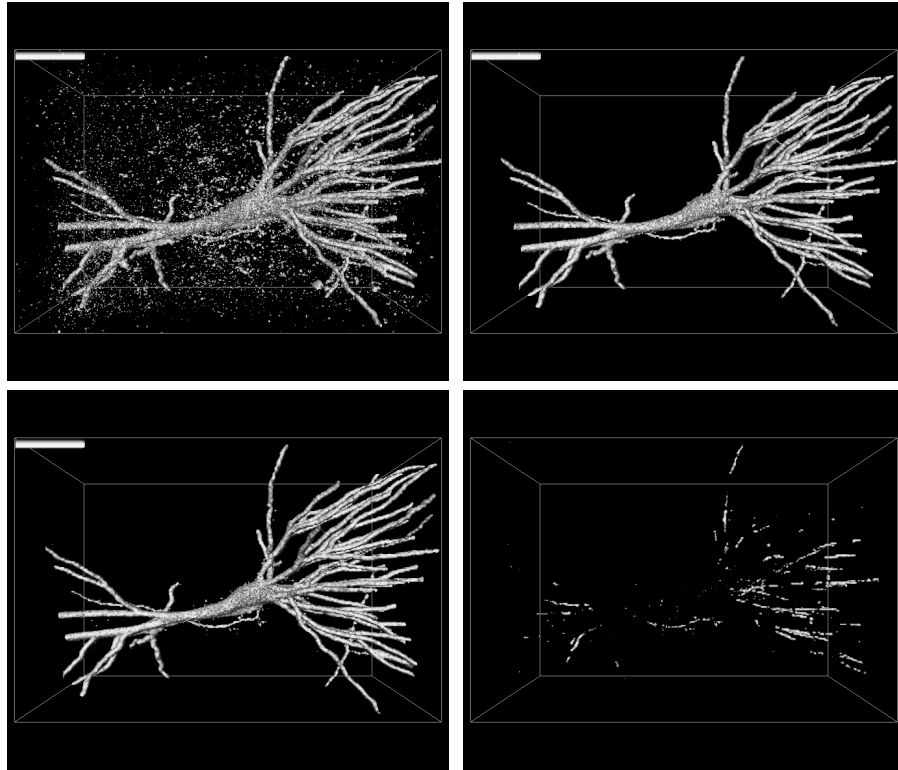


Figure 4.1: Isosurface projections of a confocal laser scanning micrograph of a pyramidal neuron and the output of the non-compactness filter (4.4) based on the 26-connectivity, both at isolevel 1. The first image in the bottom row illustrates the filter's performance using closing-based connectivity and the second shows the difference volumes between two attribute filter results. Various details within the neuron are lost using the 26-connectivity which are preserved by using a second-generation connectivity instead. See [57] for details.

can be found in [57]. Furthermore, an investigation in optimizing the parameters affecting the performance of these filters is discussed in [56] An example of attribute thinnings using closing-based second-generation connectivity is shown in Figure 4.1.

4.3 The Max-Tree algorithm

The Max-Tree was introduced by Salembier [65] as a versatile structure for computing anti-extensive attribute filters on images and video sequences. It is a rooted, unidirected tree

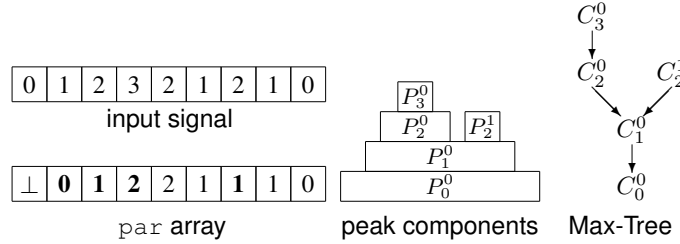


Figure 4.2: Example of input signal, peak components, Max-Tree and its encoding in a *par* array, in which \perp denotes the overall root node, and boldface numbers denote the level roots, i.e., they point to positions in the input with gray level other than their own.

in which the node hierarchy corresponds to the nesting of peak components given a gray-scale image. A peak component P_h at level h is a connected component of the thresholded image $T_h(f)$. Each tree node C_h^k (k is the node index) contains only those pixels of a given peak component which have gray-level h . In addition each node except for the root, points towards its parent $C_{h'}^{k'}$ with $h' < h$. The root node is defined at the minimum level h_{min} and contains the set of pixels belonging to the background.

The algorithm is a three-stage process in which the construction of the tree and the computation of node attributes is independent of filtering and image restitution. During the construction stage every pixel visited contributes to the auxiliary data buffer associated to the node it belongs to. Once a node is finalized, its parent inherits these data and re-computes its attribute. Inheritance in the case of increasing attributes such as area/volume is a simple addition while for non-increasing attributes such as the non-compactness measure of (4.4) the accumulation relies on more delicate attribute handling functions described in [57].

4.4 Including union-find in the Max-Tree

The hierarchical queue-based algorithm given by Salembier [65] cannot be trivially parallelized. In our approach we choose to partition the image into N_p connected disjoint regions the union of which is the entire image domain. Each region is assigned to one of the N_p processors for which a separate tree is constructed. The non-trivial part of this approach is the merging of the resulting trees. It is a process that requires (i) the merging of the peak components P_h^i , (ii) the updating of the parent relationships, and (iii) the merging of the attributes of the peak components. Parallellizing the filtering stage is trivial.

Previously, Najman et al. provided an algorithm to compute the Max-Tree using union-find [52]. Wilkinson et al [93] use a different approach, using Salembier et al's original algorithm [65] and changing the way the labels indicating node-membership of each pixel were chosen. Instead of using arbitrary numbers, Wilkinson et al use the index of the first

pixel of a node as the label. This means that each pixel of a node points to this “canonical element”, which is referred to as a *level root*. The level root of a node itself is given the level root of its parent node as its index. These labels (or actually parent pointers in union-find terms) are stored in an array denoted `par`. Thus, if $f(\text{par}[x]) \neq f(x)$, x is a level root. In the algorithm in [93], after building a tree using a single thread, each `par[x]` points directly to a level root: its own if x is not a level root, or to the level root of the parent node. An example is shown in Figure 4.2. Once the results of multiple threads are merged, this is no longer true. Therefore, we implement a function `levroot` to find the level root of any pixel. If `levroot(x) = levroot(y)` x and y belong to the same node. The implementation of `levroot` also includes path compression as in [79].

4.5 The dual-input mode

As in the sequential case, the structure of the Max-Tree is dictated by the peak components of the mask volume m rather than the original volume f . An example is given in Figure 4.3. The dual-input version of the algorithm in [93] requires a number dummy nodes which assist in the merging of the different trees once all the threads return. To do this we double the size of the `par` array, and place the volumes f and m side by side in a single block of memory. In this way $f(p + \text{volsize}) = m(p)$ for all voxels p in the volume domain. For all p for which $f(p) \neq m(p)$ `par(p + volsize)` will contain a valid reference to a level root.

The flooding function proceeds as described in [93] only we modify the way auxiliary data are handled and add a number of intensity mismatch checks to conform with the dual-input algorithm. After reaching a given level lev (=current level in mask m) and before retrieving any of the pixels available in the queue for that level, we first initialize the auxiliary data variable `attr`. It is set to the attribute count of the node corresponding to the `lero[lev]`. If an attribute count from a node at higher level is inherited through parameter `thisattr`, we update `attr`. A while loop then retrieves sequentially the members of the queue and for each one performs the mismatch check. If $f(p) \neq m(p)$ for a pixel p this signals the case in which p belongs to the current active node at $f(p)$ through the connected component at level $m(p)$, i.e. it defines a peak component at level $f(p)$ to which p in the mask volume is connected. In terms of our parallelizing strategy this means that it already defines a dummy node at $m(p)$ offset by `volsize`. We must then set `par(p + volsize)` to `lero[lev]`. We must also create a new node at level $f(p)$ if none exists, and add p to the node at level $f(p)$. If $f(p) > m(p)$ p is a singleton (according to (4.1)). This requires finalizing the node which is done by setting its parent to `lero[lev]`, setting its auxiliary data to the unit measure and clearing `lero[f(p)]`. Details are given in Algorithm 1.

Otherwise, if $f(p) = m(p)$, it is necessary to check if the `lero[lev] ≥ volsize`, i.e. if it is a dummy node. If this is the case, we update `par[lero[lev]]` to p , and then set `lero[lev]` to p , effectively setting the level root to a non-dummy node. The auxiliary data stored in `attr`

Algorithm 1 *The flooding function of the concurrent Dual-Input Max-Tree algorithm.*

```

procedure LocalTreeFlood(threadno, lero, lev, thisattr) =
  Initialize auxilliary attribute data attr and merge with thisattr
  while (QueueNotEmpty(set, lev)) do
    retrieve p from queue at level lev
    if  $f(p) \neq lev$  then
       $par[p + volsize] := lero[lev]$ ;
      if node at level  $f(p)$  exists then
        add p to it;  $par[p] := lero[f(p)]$ ;
      else
        create node at level  $f(p)$ ;  $lero[f(p)] := p$ ;
      end;
      if  $f(p) > lev$  then (* singleton with parent at lev *)
        finalize node; add p to attr;  $par[p] := lero[lev]$ ;
      end;
    else (* No mismatch *)
      if  $lero[lev] \geq volsize$  then (* First pixel at level lev *)
         $par[lero[lev]] := p$ ;  $lero[lev] := p$ ;
      end;
      add p to attr;
    end; (* No mismatch *)
  end; (* while *)
  for all neighbours q of p do
    if not processed[q] then
      processed[q] := true;  $m_q := m(q)$ ;
      initialize childattr to empty;
      if  $m(q) \neq f(q)$  then newnode :=  $q + volsize$ ;
      else newnode := q; end;
      if lero[ $m(q)$ ] does not exist then lero[ $m(q)$ ] := newnode;
      else  $par[newnode] := lero[m(q)]$ ; end;
      while  $m_q > lev$  do
         $m_q := LocalTreeFlood(threadno, lero, m_q, childattr)$ ;
      end;
      add any data in childattr to attr;
    end;
  end; (* for *)
  detect parent of lero[lev]
  add auxilliary data in attr to auxilliary data of lero[lev]
  set thisattr to attribute data of lero[lev]
  return level of parent of lero[lev]
end LocalTreeFlood.

```

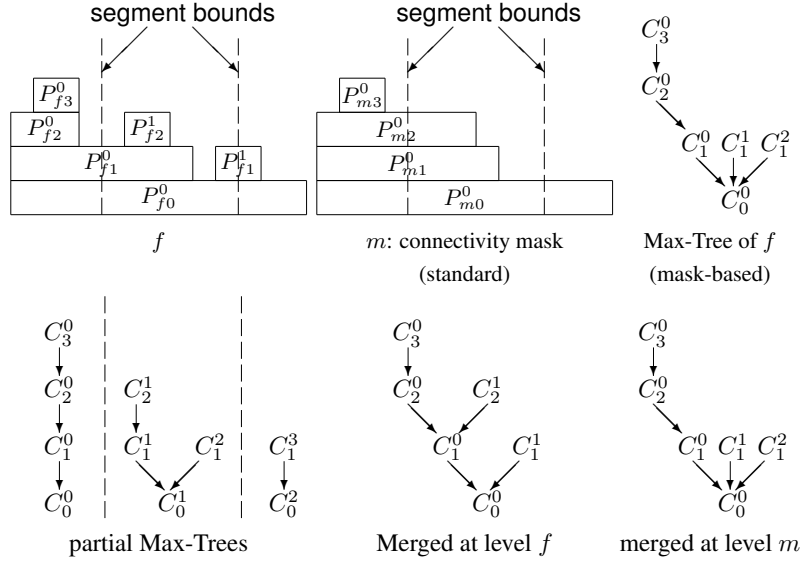


Figure 4.3: Dual Input Max-Tree of 1-D signal f using mask m : The attributes of C_2^0 and C_2^1 are merged to C_2^0 since all pixels at level $h = 2$ are clustered to a single peak component. Furthermore C_1^1 breaks up to a number of singleton nodes equal to the number of pixels in P_{f1}^1 . Bottom row: partial Max-Trees of segments of signal indicated by the dashed lines; merger of partial Max-Trees at level of f at the boundaries yields standard Max-Tree in this case; merging at level of m at the boundary yields correct result.

are then updated.

For every unprocessed neighbour q of p we determine where to create a new node. If $f(q) = m(q)$ the new node is q , otherwise $q + volsize$. If $lero[m(q)]$ exists, we set $par[neunode]$ to $lero[m(q)]$, otherwise $lero[m(q)]$ is set to $par[neunode]$. If $m(q) \geq lev$ we then enter into the recursion as in [65, 93].

4.6 Concurrent merging of Max-Trees

As in regular connectivities, we must now connect the N_p Max-Trees. In [93], this is done by inspecting the pixels along the boundary between the parts, and performing the `connect` function on adjacent pixels on either side of the boundary. This function is shown in Algorithm 3. A proof of the correctness and a detailed discussion are given in [93]. The key reason why this works efficiently, is that merging two nodes containing x and y , with $f(x) = f(y)$

Algorithm 2 *Concurrent construction and filtering of the Max-Trees, thread p .*

```

process ccaf( $p$ )
  build dual input Max-Tree  $Tree(p)$  for segment belonging to  $p$ 
  var  $i := 1, q := p$ ;
  while  $p + i < K \wedge q \bmod 2 = 0$  do
    wait to glue with right-hand neighbor ;
    for all edges  $(x, y)$  between  $Tree(p)$  and  $Tree(p + i)$  do
      if  $f(x) \neq m(x)$  then  $x := x + volsize$ ;
      if  $f(y) \neq m(y)$  then  $y := y + volsize$ ;
      connect( $x, y$ );
    end ;
     $i := 2 * i; q := q / 2$ ;
  end ;
  if  $p = 0$  then
    release the waiting threads
  else
    signal left-hand neighbor ;
    wait for thread 0
  end ;
  filter( $p, lambda$ );
end ccaf.

```

reduces to the assignment

$$\text{par}[\text{levroot}(y)] := \text{levroot}(x). \quad (4.5)$$

This is easily verified as follows: $\text{par}[\text{levroot}(y)]$ now points to a pixel with the same grey level because $f(x) = f(y)$, and $\text{levroot}(x) = \text{levroot}(y)$ after assignment (4.5), so that x and y belong to the same node.

Function `connect` is called by the process *concurrent construction and filter* or `ccaf` (see Algorithm 2), which corresponds to one of the threads of the concurrent merging algorithm. Each thread p first builds a Max-Tree for its own sub-domain V_p .

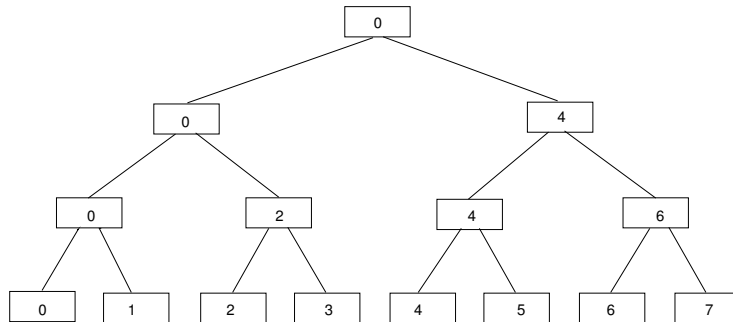
Process `ccaf` is called after initializing `par`, the auxiliary data functions and preparing the thread data. It starts off by first initializing the level root array `lero` and hierarchical queue for all gray-levels and finding the minimum voxel values in f and m . Having got the starting voxel of minimum grey value in m it calls `LocalTreeFlood`. If the minimum values in f and m differ, some post-processing as explained in [57] is required.

Algorithm 3 *Merging two Max-Trees*

```

procedure connect( $x, y$ ) =
  Initialize auxilliary attribute data temp1 to empty
   $x := \text{levroot}(x)$ ;  $y := \text{levroot}(y)$ ;
  if  $f(y) > f(x)$  then swap( $x, y$ ) end
  while  $x \neq y \wedge y \neq \perp$  do
     $z := \text{levroot}(\text{par}[x])$ 
    if  $z \neq \perp \wedge f(z) \geq f(y)$  then
      Add data in temp1 to attribute data of  $x$ ;
       $x := z$ ;
    else
      temp2 := sum of attribute data of  $x$  and temp1;
      temp1 := attribute data of  $x$ ;
      attribute data of  $x := \text{temp2}$ ;
       $\text{par}[x] := y$ ;  $x := y$ ;  $y := z$ ;
    end
  end
  if  $y = \perp$  then (* root of one tree reached *)
    while  $x \neq \perp$  do (* process remaining ancestors of  $x$  *)
      Add data in temp1 to attribute data of  $x$ ;
       $x := \text{levroot}(\text{par}[x])$ ;
    end
  end
end connect.

```

**Figure 4.4:** Binary tree used for merging domains.

After this, the sub-domains are merged by means of a binary tree in which thread p accepts all sub-domains V_{p+i} with $p+i < N_p$ and $0 \leq i < 2^a$, where 2^a is the largest power of 2 that divides p . An example of a binary tree for $N_p = 8$ is shown in Figure 4.4. Note that odd-numbered threads accept no sub-domains. A thread that needs to accept the domain of its right-hand neighbor, has to wait until the neighbor has completed its Max-Tree computation. Because the final combination is computed by thread 0, all other threads must wait for thread 0 before they can resume their computation for the filtering phase. This synchronization is realized by means of two arrays of $N_p - 1$ binary semaphores. The filtering phase is also fully concurrent, and is identical to that described in [93].

For second-generation connectivity, the difference lies not in the implementation of `connect`, but in *which* pixels need to be merged. Suppose x and y are adjacent voxels which lie on different sides of the boundary inspected by `ccaf`. If $f(x) = m(x)$ the node in the Max-Tree at level $f(x)$ is the correct one, as before, otherwise we should start merging at level $m(x)$, as shown in Figure 4.3. At the left-hand segment boundary in this figure, merging at level $f(x)$ ignores the fact that $P_{f_2}^0$ and $P_{f_2}^1$ are clustered together in node C_2^0 using connectivity based on mask m . By contrast, at the right-hand segment boundary, merging from level $f(x)$ would merge nodes C_1^2 and C_1^3 , which are considered singletons in the mask-based connectivity. In the scheme outlined above, this means that we start the merger from x if $f(x) = m(x)$, and from $x + volsize$, otherwise. The same holds for y . Thus the only changes to the `ccaf` function when compared to [93] lies in the statements immediately preceding the call to `connect`.

4.7 Performance testing and complexity

The above algorithm was implemented in C for the general class of anti-extensive attribute filters. Wall-clock run times for numbers of threads equal to 1, 2, 4, 8, 16, 32, and 64 for two different attributes were determined. The attributes chosen were volume (yielding an attribute opening) and the non-compactness measure (4.4) [94] yielding an attribute thinning.

Timings were performed on an AMD dual-core, Opteron-based machine. This machine has two dual-core Opteron 280 processors at 2.4 GHz, giving a total of 4 processor cores, and 8 GB of memory (4 GB per processor socket). Each timing was repeated 10 times, and the minimum was used as the most representative of the algorithm's performance. Five volume data sets publicly available from <http://www.volvis.org> were used. All volumes were 8 bit/voxel sets, comprising 4 CT-scans and 1 MRI scan. Tests were done using volume openings with $\lambda = 100$ and φ_1 with $\lambda = 2.0$ and the subtractive rule. The volume sizes ranged from 22.7 to 128 MB. The speed-up achieved is shown in Figure 4.5. As can be seen, the speed-up is slightly better than linear, as we move from 1 to 4 threads (4.21 ± 0.15 for volume openings and 4.14 ± 0.15 for non-compactness thinning at 4 threads). This may be due to the fact that more than 4 GB of memory is required when processing the larger

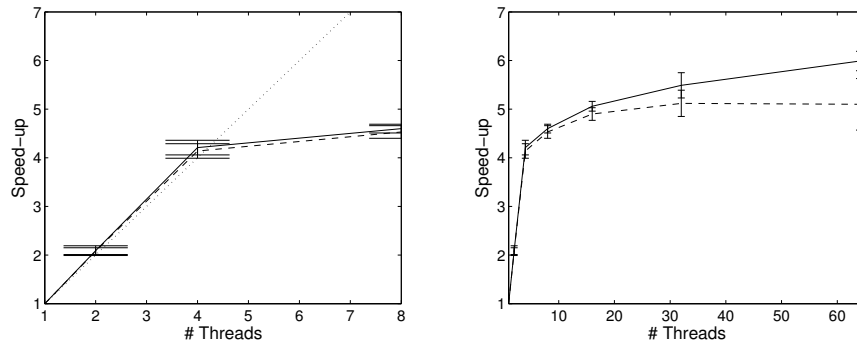


Figure 4.5: Speed-up for volume openings (solid) and non-compactness thinnings (dashed) as a function of number of threads. The left graph shows the initial, slightly better than linear (dotted-line) speed-up as we move from 1 to 4 threads. The right-hand graph also shows the behavior up to 64 threads.

volumes in the set, and therefore the processor doing the work requires access to the memory bank of the other socket, resulting in higher latency. As the number of threads exceeds the number of cores, we still obtain more speed-up, up to 5.99 ± 0.2 at 64 threads for volume openings, and 5.12 ± 0.27 at 32 threads for non-compactness thinning. In absolute terms, computing time went from between 20.8 and 128 s down to between 4.66 and 23.4 s.

The complexity of the algorithm is governed by two main parts: the building phase and the merging phase. Assuming a volume of $X \times Y \times Z = N$, in the building phase the time complexity is $O(GN/N_p)$, with G the number of grey levels, and N_p the number of processors. This complexity arises from the $O(GN)$ complexity of Salembier et al’s Max-Tree algorithm [47]. If the number of grey levels is large, it may be better to replace this by Najman and Courpie’s method [52]. The merging phase has complexity $O(GXY \log N \log N_p)$ if the volume is split up into slices orthogonal to the Z direction. The $\log N$ is due to the fact that we only use path compression, not union-by-rank. Memory requirements are $O(N+G)$.

4.8 Conclusions

The speed-up of the algorithm presented is similar to that of the parallel Max-Tree algorithm in [93]. However, it is about 50% slower in absolute terms. The speed-up if the number of threads exceeds the number of physical processors is due to reduced cache thrashing, as is confirmed by profiling. It also indicates that on machines with more processing cores, a (near) linear speed up beyond 4 CPUs is expected.

Apart from use in 3-D data, the algorithm could be of use in the efficient implementation

of attribute-space connected filters [92], in which the 2-D input image is embedded into a higher-dimensional attribute space, followed by application of a connected filter in that space.

Given the ready availability of multi-core processors, this algorithm is not restricted to supercomputers anymore, but will be of use to many, and in the near future most desktop machines.

Chapter 5

Partition-Induced Connections and Operators for Pattern Analysis

Everything has its beauty but not everyone sees it.

Confucius

Abstract

In this paper we present a generalization on the notion of image connectivity similar to that modeled by second-generation connections. The connected operators based on this new type of connection make use of image partitions aided by mask images to extract pathwise connected regions that were previously treated as sets of singletons. This leads to a redistribution of image power which affects texture descriptors. These operators find applications in problems involving contraction-based connectivities, and we show how they can be used to counter the oversegmentation problem reported in literature. Despite restrictions which prevent extensions to gray-scale, we present a method for gray-scale spectral analysis of biomedical images characterized by filamentous details. Using connected pattern spectra as feature vectors to train a classifier we show that the new operators outperform the existing contraction-based ones and that the classification performance competes with, and in some cases outperforms methods based on the standard 4- or 8- connectivity. Finally, combining the two methods we enrich the texture description and increase the overall classification rate.

5.1 Introduction

IN image analysis it is often desirable to sort objects based on their structural characteristics, typically expressed by means of some attribute measure. The pattern spectrum [41] is a commonly used method that features ordered attribute classes that keep track of the amount of image detail or power (measured in number of pixels) that falls within their range of attribute values.

Pattern spectra can be computed from granulometries [11, 29, 47, 65, 66, 68, 74] which are ordered sets of morphological operators adhering to some properties discussed later. The operators can either be structural or connected filters, each allowing a limited range of image

details to pass. A recent comparison between the two filter types favored granulometries based on connected filters [85]. The method presented for computing pattern spectra, compared with other existing methods was shown to be rotation invariant and significantly less sensitive to noise, to allow for multi-dimensional spectra to be computed based on strict size and shape attributes and its computation time was independent of the number of scales or shape classes being used.

Connected filters are shape preserving operators [29, 62, 68] which work by removing or retaining connected image regions known as *connected components*, but without introducing new ones. If filtering is based on the attribute value of the examined connected component they are referred to as *attribute* or *grain filters* [11, 29, 66]. Such filters have been used among other areas, in biomedical imaging for filament enhancement and area/volume filtering [86, 90, 94].

Connected operators rely on some notion of connectivity, commonly the 4 and 8 pixel adjacency relations [37]. With connectivity expressed in a set-theoretic framework [45, 68] several generalizations were introduced which overcame topological constraints of earlier formalisms. An example is the *second-generation connectivity* [7, 10, 57, 62, 69, 70] in which operators associated with it can access families of sets that account for connected components that are not strictly 4 or 8 connected. Typically we refer to clusters or sub-regions of connected components according to the classical connectivity or for combinations of the two. An efficient scheme allowing for all three cases, has been recently introduced and is termed *mask-based connectivity* [57]. According to this, the connectivity of an image given the standard 4 and 8 connection can be obtained by a second image (the *connectivity mask*) that commonly results from the application of some operator on the original. This allows an image to be connected in any arbitrary way. Like earlier formulations, connectivity openings associated with mask-based connectivity extract the connected components of interest and handle the remaining structures as singleton sets. Generating singletons is a feature that has been used to counter the *leakage* problem of connected operators [65]. The leakage results from thin elongated paths connecting different objects in an image that should be treated individually and is usually caused by background texture, noise or other image details. The treatment of these paths as groups of singleton sets is known under certain conditions to generate problems in both filtering and segmentation and examples referred to as *oversegmentation* or *fragmentation* are addressed in [54, 92].

In this paper we counter the problem of oversegmentation by introducing a more general connectivity scheme that stems from multiple partitions of a given image aided by a connectivity mask (Section 5.3). The associated connected operators instead of generating singletons, consider pathwise connected elements of the remaining structures as individual connected components. This is shown to limit their extensions to gray-scale (Section 5.3.4) and thus affects the way pattern spectra can be computed. We formalize the Max-Tree method from [85] in a statement (Section 5.4) from which we derive a way for computing

gray-scale connected pattern spectra using this new type of texture information. To demonstrate the potential of this new class of connected operators we experiment with texture based classification of diatom images using the spectrum as a feature vector and compare the performance with that achieved using pattern spectra based on standard and contraction-based second-generation connectivities (Section 5.5).

5.2 Theory

5.2.1 Connections and Connected Operators

In mathematical morphology the concept of connectivity is defined by the notion of *connectivity classes* [45, 68]:

Definition 13. *Let E be an arbitrary space. A connectivity class or connection \mathcal{C} is any family in $\mathcal{P}(E)$ that satisfies:*

1. $\emptyset \in \mathcal{C}$ and for all $x \in E$, $\{x\} \in \mathcal{C}$,
2. for any $\{A_i\} \subseteq \mathcal{C}$ for which $\bigcap A_i \neq \emptyset \Rightarrow \bigcup A_i \in \mathcal{C}$

This means that both the empty set and singleton sets, denoted as $\{x\}$, are connected, and any union of elements of \mathcal{C} which have a non-empty intersection is also connected. The members of \mathcal{C} are called *connected sets* and are element groupings of E .

Every set $X \subseteq E$ can be written as the union of pairwise disjoint connected sets of maximal extent, C_i . Maximality in this sense means that given a set C_i there can be no other set $C_j \supset C_i$ such that $C_j \subseteq X$ and $C_j \in \mathcal{C}$. A set $C_i \subseteq X$ also called a *connected component* or a *grain* of X , given a point $x \in C_i$ is addressed by a *connectivity opening* which is an operator defined as:

$$\Gamma_x(X) = \bigcup \{A_i \in \mathcal{C} \mid x \in A_i \text{ and } A_i \subseteq X\}. \quad (5.1)$$

With all A_i containing x in their intersection, their union $\Gamma_x(X)$ is also connected and furthermore, $\Gamma_x(X) = \emptyset$ if $x \notin X$.

Connectivity openings are algebraic openings and therefore are *anti-extensive*, *increasing* and *idempotent* operators. For any set X each property implies the following:

1. Anti-extensivity: $\Gamma_x(X) \subseteq X$,
2. Increasingness: if $X \subseteq Y \Rightarrow \Gamma_x(X) \subseteq \Gamma_x(Y)$,
3. Idempotence: $\Gamma_x(\Gamma_x(X)) = \Gamma_x(X)$.

The operator Γ_x is explicitly related to a connectivity class \mathcal{C} if satisfying the set of conditions given by Serra [68] (also in [29, 62]) in the following theorem:

Theorem 3. *The datum of a connectivity class \mathcal{C} on $\mathcal{P}(E)$ is equivalent to the family $\{\Gamma_x \mid x \in E\}$ of openings on x such that:*

1. every Γ_x is an algebraic opening,
2. for all $x \in E$, we have $\Gamma_x(X) = \{x\}$,
3. for all $X \subseteq E$, $x, y \in E$, $\Gamma_x(X)$ and $\Gamma_y(X)$ are equal or disjoint,
4. for all $X \subseteq E$, and all $x \in E$, we have $x \notin X \Rightarrow \Gamma_x(X) = \emptyset$.

Concluding, it can be seen that connectivity openings characterize uniquely the connectivity class they are associated with and there is a one-to-one correspondence between the two.

5.2.2 Second-Generation Connectivity

The definition of connectivity by means of connectivity classes allows several generalizations. Second-generation connectivity is such an example where from a given "parent" class \mathcal{C} we derive a "child" class given some image transformation captured by the corresponding connectivity opening. This concept is modeled by two types of connections, the *clustering* and *contraction*-based connectivity classes.

When clustering, disconnected components according to \mathcal{C} satisfying some structural criteria, most commonly the distance separating them, are extracted as a single entity. By contrast, in a contractive transformation object regions that fail some structural criteria, most commonly the local width, are converted to singletons, splitting wide object regions connected by narrow bridges apart this way. The two connectivity transformations can be combined in a single framework known as *mask-based* or *m-connectivity* [57] in which the grains of a mask image M are used to selectively carry out clusterings or contractions on the connected components from the original.

A mask-based connectivity class is defined as follows:

$$\mathcal{C}^M = \{\emptyset\} \cup \mathcal{S} \cup \{A \subseteq E \mid \exists x \in E : A \subseteq \Gamma_x(M)\} \quad (5.2)$$

and a connectivity opening from the corresponding family $\{\Gamma_x^M(X) \mid x \in E\}$ as:

$$\Gamma_x^M(X) = \begin{cases} \Gamma_x(M) \cap X & \text{if } x \in X \cap M, \\ \{x\} & \text{if } x \in X \setminus M, \\ \emptyset & \text{otherwise.} \end{cases} \quad (5.3a)$$

$$(5.3b)$$

$$(5.3c)$$

The family of operators Γ_x^M essentially "masks" the desired members of \mathcal{C} to \mathcal{C}^M by selecting all subsets of X found within the grains of M . An important feature of the definition above is that there are no assumptions as to how M should be generated. This eliminates constraints in the ways the image domain can be connected.

5.2.3 Attribute Filters

The notion of a connected filter in mathematical morphology describes a mapping $\psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ that is increasing and idempotent [28, 29, 68]. The connectivity opening is a trivial example and based on it we can define a number of other connected filters that work by imposing constraints on the connected components it returns. Constraints are commonly expressed in the form of attribute criteria to accept or to reject connected components based on some attribute measure. Attribute criteria Λ are put in place by means of a trivial opening Γ_Λ . The later is defined as an operator $\Gamma_\Lambda : \mathcal{C} \rightarrow \mathcal{C}$ which if applied on a connected component $C \in \mathcal{C}$ yields C if $\Lambda(C)$ is true, and \emptyset otherwise. Obviously, $\Gamma_\Lambda(\emptyset) = \emptyset$. Attribute criteria are often expressed as:

$$\Lambda(C) = Attr(C) \geq \lambda \quad (5.4)$$

with $Attr(C)$ some real-valued attribute of C , and λ an attribute threshold.

Definition 14. *The binary attribute opening Γ^Λ of a set X with an increasing criterion Λ is given by:*

$$\Gamma^\Lambda(X) = \bigcup_{x \in X} \Gamma_\Lambda(\Gamma_x(X)) \quad (5.5)$$

An example is the area opening [14, 87].

Attribute-based connected operators may also be based on shape criteria rather than size. They are generally non-increasing operators which are scale, rotation and translation invariant. A shape operator that is also idempotent defines a *shape filter* and an example is the *attribute thinning* Φ^Λ [11, 29]. An example of a shape criterion is the non-compactness (also referred to as elongation) criterion [86, 94] given by:

$$Attr(C) = I(C)/A^2(C). \quad (5.6)$$

$I(C)$ is the moment of inertia and $A(C)$ the area of a component C .

Attribute filters can be applied on sets characterized by some generalized notion of connectivity by replacing Γ_x in (5.5) with the appropriate connectivity opening, e.g., in the mask-based second-generation case by Γ_x^M from (5.3). For cases involving the handling of contractions, such filters present a drawback known as *oversegmentation* [54, 92]. It has been shown that an attribute opening using a contraction-based connectivity [7] reduces to performing the standard attribute opening on M , unless the criterion has been set such that Γ^Λ is the identity operator [92]. This is summarized into the following:

Theorem 4. *The attribute opening Γ_M^Λ for a contraction-based connectivity with an increasing, shift invariant criterion Λ is*

$$\Gamma_M^\Lambda(X) = \begin{cases} X & \text{if } \Lambda(\{x\}) \text{ is true} \\ \Gamma^\Lambda(M) & \text{otherwise} \end{cases} \quad (5.7a)$$

$$(5.7b)$$

where Γ^Λ is the underlying attribute opening from (5.5). It is evident that if Γ^Λ is not the identity operator, then all the singleton sets generated by the connectivity opening of (5.3) fail the attribute criterion hence filtering X reduces to filtering M instead. Oversegmentation affects any region of X not overlapping with a grain of M and applies equally to attribute thinnings.

5.2.4 Granulometries and Pattern Spectra

Attribute filters are used among other areas for constructing granulometries and computing pattern spectra [11, 65, 89].

Definition 15. *A binary size granulometry is a set of operators $\{\Gamma_r\}$ with r from some totally ordered set Λ , with the following three properties*

$$\Gamma_r(X) \subseteq X \quad (5.8)$$

$$X \subseteq Y \Rightarrow \Gamma_r(X) \subseteq \Gamma_r(Y) \quad (5.9)$$

$$\Gamma_r(\Gamma_s(X)) = \Gamma_{\max(r,s)}(X), \quad (5.10)$$

for all $r, s \in \Lambda$.

The first two properties state that Γ_r is anti-extensive and increasing, and the third implies idempotence. This summarizes essentially the definition of a size granulometry to a set of openings. The pattern spectrum $s_\Gamma(X)$ obtained by applying the size granulometry $\{\Gamma_r\}$ to a binary image X is defined as

$$(s_\Gamma(X))(u) = - \left. \frac{d\xi(\Gamma_r(X))}{dr} \right|_{r=u} \quad (5.11)$$

where ξ denotes the Lebesgue measure in \mathbb{R}^n which is simply the area $A(X)$ for $n = 2$.

Shape operators insensitive to size information are also used to define granulometries [85]. This requires omitting the second property of Def. 15 and instead include a condition ensuring scale invariance as follows

$$\Phi_r(tX) = t(\Phi_r(X)), \quad \forall t > 0. \quad (5.12)$$

Thus a shape granulometry consists of operators Φ which are anti-extensive, idempotent and scale invariant. Furthermore, shape pattern spectra can be defined in a way analogous to size pattern spectra [85].

5.3 Partition-Induced Connections

5.3.1 Partitions and Connections

The notion of a partition like that of a connection, describes element groupings on E . The formal definition as given in [70] is the following:

Definition 16. *Let E be an arbitrary set. A partition \mathbf{P} of E is a mapping $x \rightarrow \mathbf{P}(x)$ from E into $\mathcal{P}(E)$ such that*

1. *for all $x \in E : x \in \mathbf{P}(x)$,*
2. *for all $x, y \in E : \mathbf{P}(x) = \mathbf{P}(y)$ or $\mathbf{P}(x) \cap \mathbf{P}(y) = \emptyset$.*

$\mathbf{P}(x)$ is called the class of the partition of origin x . The two conditions indicate that classes $\mathbf{P}(x)$ occupy the whole space E and that two distinct classes have no common point.

Partition classes as opposed to connected components, do not necessary contain elements from the foreground sets only. Because of this, establishing a relation with a connection requires the use of connectivity openings which naturally separate background from foreground components [68, 72].

Definition 17. *Given a partition \mathbf{P} of the space E , all the subsets of each class $\mathbf{P}(x)$, $x \in E$, of the partition generate a family conditionally closed under union given by*

$$\mathcal{C}^\pi = \{A \cap \mathbf{P}(x), x \in E \text{ and } A \in \mathcal{P}(E)\}. \quad (5.13)$$

We call \mathcal{C}^π a *partition-induced* (pi) or π -connection and the associated operators, π -connectivity openings. It follows that for a set $A \subseteq E$, the connected component given by $\Gamma_x^\pi(A)$ is simply :

$$\Gamma_x^\pi(A) = A \cap \mathbf{P}(x). \quad (5.14)$$

Serra [72] concludes with the following theorem linking the notion of a partition with that of connection.

Theorem 5. *Let \mathcal{C}^π be a connection on $\mathcal{P}(E)$ associated to the family of connectivity openings $\{\Gamma_x^\pi \mid x \in E\}$. For each set $A \subseteq E$ the connectivity openings $\{\Gamma_x^\pi \mid x \in E\}$ subdivide A according to the largest possible partition into members of \mathcal{C}^π . This operation is increasing in that if $A \subseteq A'$, then any connected component of A is upper-bounded by a connected component of A' .*

5.3.2 Countering Oversegmentation with π -Connections

The connectivity openings associated to contraction-based or mask-based second-generation connections return singleton sets that account for foreground elements of the original set X that correspond to the background in the connectivity mask M or $\psi(X)$ (where ψ typically an erosion or an opening). Attribute filters based on such connectivity openings yield oversegmented sets as discussed in Section 5.2.3 and furthermore disregard structural information from objects in regions given by $X \setminus M$. In this section, aided by the concept of partitions, we introduce a connectivity opening that addresses elements in these regions as connected components thus allowing to assign to them meaningful attributes and process them further.

Consider a partition of E such that given any arbitrary set $A \subseteq E$, its classes are given by:

$$\mathbf{P}_A(x) = \begin{cases} \Gamma_x(A) & \text{if } x \in A, \\ \{x\} & \text{otherwise,} \end{cases} \quad (5.15a)$$

$$(5.15b)$$

The proof that \mathbf{P}_A is a valid partition is trivial. Given a mask image M resulting from some operator applied on X , substituting $\mathbf{P}_M(x)$ in (5.14) yields:

$$\Gamma_x^\pi(X) = X \cap P_M(x) = \begin{cases} \Gamma_x(M) \cap X & \text{if } x \in X \cap M, \\ \{x\} & \text{if } x \in X \setminus M, \\ \emptyset & \text{otherwise.} \end{cases} \quad (5.16a)$$

$$(5.16b)$$

$$(5.16c)$$

This is the mask-based connectivity opening discussed in Section 5.2.2, derived in a far simpler way than in [57]. The objective is to replace the term returning singleton sets with a more specific function to extract components in $X \setminus M$.

Proposition 3. *Let \mathcal{C} be a connection of E associated with the family $\{\Gamma_x \mid x \in E\}$ of connectivity openings. The mapping of $x \rightarrow \mathbf{P}_M^X(x)$ from E to $\mathcal{P}(E)$ is a partition whose classes are given by:*

$$\mathbf{P}_M^X(x) = \begin{cases} \mathbf{P}_M(x) & \text{if } x \in M, \\ \mathbf{P}_{M^c}(x) \cap \mathbf{P}_X(x) & \text{otherwise,} \end{cases} \quad (5.17a)$$

$$(5.17b)$$

where M^c is the complement of the mask image M .

Proof First we show that the classes of \mathbf{P}_M^X cover E , i.e. $\bigcup_{x \in E} \mathbf{P}_M^X(x) = E$.

$$\bigcup_{x \in E} \mathbf{P}_M^X(x) = \left(\bigcup_{x \in M} \mathbf{P}_M^X(x) \right) \cup \left(\bigcup_{x \notin M} \mathbf{P}_M^X(x) \right) \quad (5.18)$$

The first term is trivial since $\bigcup_{x \in M} \mathbf{P}_M^X(x) = \bigcup_{x \in M} \Gamma_x(M) = M$. For the second term we identify two subcases:

$$\bigcup_{x \notin M} \mathbf{P}_M^X(x) = \left(\bigcup_{x \in X \setminus M} \mathbf{P}_M^X(x) \right) \cup \left(\bigcup_{x \notin X \cup M} \mathbf{P}_M^X(x) \right). \quad (5.19)$$

Substituting (5.15) with the appropriate subscript we get $\bigcup_{x \in X \setminus M} \Gamma_x(M^c) \cap \Gamma_x(X) = \bigcup_{x \in X \setminus M} \Gamma_x(X \setminus M) = X \setminus M$ for the first, and $\bigcup_{x \notin X \cup M} \Gamma_x(M^c) \cap \{x\} = \bigcup_{x \notin X \cup M} \{x\} = E \setminus X \cup M$ for the second subcase. Summarizing, (5.18) yields $X \setminus M \cup M \cup \{\{x\} \mid x \notin X \cup M\} = E$.

For the last part of the proof we are required to show that the classes of the partition \mathbf{P}_M^X are equal or disjoint; that is for any two points of origin $x, y \in E \Rightarrow \mathbf{P}_M^X(x) = \mathbf{P}_M^X(y)$ or $\mathbf{P}_M^X(x) \cap \mathbf{P}_M^X(y) = \emptyset$. We identify the following three cases:

1. if $x, y \in M$ then either $\Gamma_x(M) = \Gamma_y(M)$ or $\Gamma_x(M) \cap \Gamma_y(M) = \emptyset$ by the definition of connectivity openings,
2. if $x \in M$ and $y \notin M$ then depending on whether $y \in X$ or not we have $\Gamma_x(M) \cap \Gamma_y(X \setminus M) = \emptyset$ because M and $X \setminus M$ are disjoint sets or $\Gamma_x(M) \cap \{y\} = \emptyset$ otherwise,
3. if $x, y \notin M$ then we have four subcases:
 - (a) $x, y \in X \Rightarrow \Gamma_x(X \setminus M) = \Gamma_y(X \setminus M)$,
 - (b) $x \in X, y \notin X \Rightarrow \Gamma_x(X \setminus M) \cap \{y\} = \emptyset$,
 - (c) $x, y \notin X \Rightarrow \{x\} = \{y\}$ or $\{x\} \cap \{y\} = \emptyset$. □

Thus, in all cases the we have equal or disjoint sets concluding that \mathbf{P}_M^X is a valid partition. This yields a partition-induced connection in the form of (5.13) with connectivity openings given by $\Gamma_x^\pi(X) = X \cap \mathbf{P}_M^X(x)$ or more explicitly:

$$\Gamma_x^\pi(X) = \begin{cases} \Gamma_x(M) \cap X & \text{if } x \in X \cap M, & (5.20a) \\ \Gamma_x(X \setminus M) & \text{if } x \in X \setminus M, & (5.20b) \\ \emptyset & \text{otherwise,} & (5.20c) \end{cases}$$

The classes of \mathbf{P}_M^X can also be set to occupy coarser regions. An example is by setting $\mathbf{P}_M^X(x) = \mathbf{P}_{M^c}(x), \forall x \notin M$ in which the corresponding connectivity opening returns a cluster of all regions in $X \setminus M$. For the purposes of the current work however we employ the partition \mathbf{P}_M^X as is defined in (5.17). A similar operator given with a proof that does not involve partitions was presented at an earlier paper [54]. There we stretch the reasons why operator-based second-generation connectivity cannot be used to define connectivity openings like in (5.20).

5.3.3 π -connected Attribute Filters

Attribute filters that are based on π -connectivity openings as opposed to contractive mask-based openings have the advantage of dealing with meaningful structures in regions given by $X \setminus M$. These structures given a contraction-based connectivity mask are usually thin

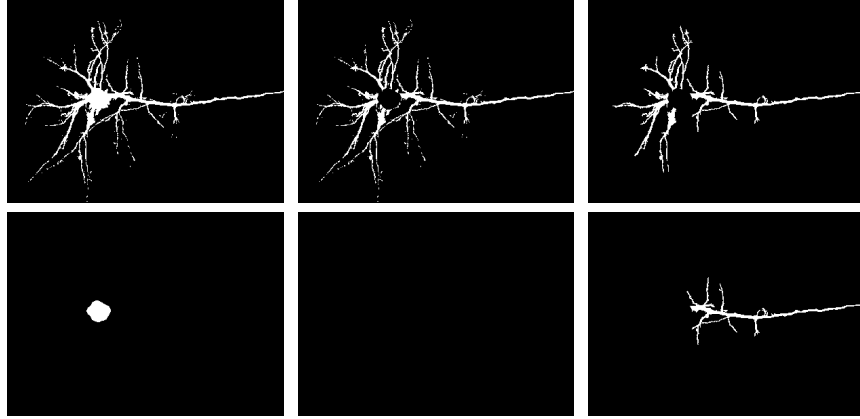


Figure 5.1: Attribute thinning of neurons: original image X (top left); the mask image M (bottom left); the image structures in $X \setminus M$ (top middle); elongation filtering of connected components in M and $X \setminus M$ with $\lambda = 5$ (bottom middle and top right respectively), and of $X \setminus M$ with $\lambda = 9$ (top right). The mask image is generated by a structural opening with a disk structuring element of radius 3.

elongated segments like the filamentous protrusions of the binary neuron image of Fig. 5.1. A filter configured with (5.3) would remove all pixels in these regions unless it is set to be the identity operator. Instead, applying an attribute thinning on the connected components as given by (5.20) removes compact structures and allows the extraction of the dendrites from the neuron soma. In this example the objective is not recovering the central object as would other methods for resolving the leakage problem do, but obtaining structural information on the "leaking" paths.

The effects of oversegmentation on similar examples given in gray-scale images are demonstrated in [54]. Thin/small structures that appear at higher gray levels often contribute to the object sharpness thus removing them causes severe blurring and edge distortion. Extending π -connected attribute filters to gray-scale is not trivial and the problem remains to be solved. Despite the limitations discussed next, π -connectivity openings/filters find use in pattern analysis of gray-scale images and provide richer spectra when compared to contraction/mask-based connectivity openings. Note that binary granulometries based on π -connectivity openings or other binary attribute filters relying on them can be trivially defined since the operator properties confirmed in the previous subsection and verified in [54], conform with the Definition 15 [85].

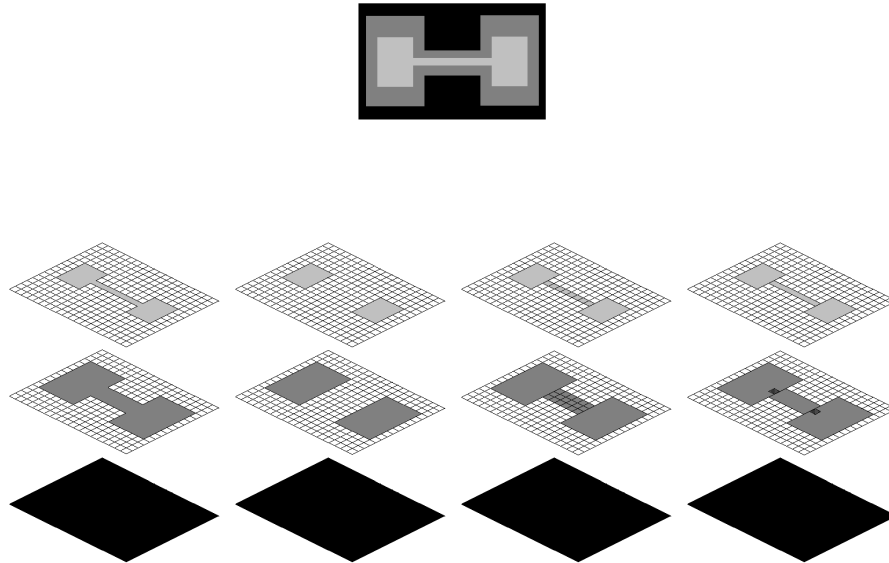


Figure 5.2: Gray-scale decompositions (from left to right): The original 3-level image f ; the mask image m by a structural opening with a square SE; f given a contraction-based m -connectivity; f given a contraction-based π -connectivity; the highlighted regions of $T_h(f)$ on the middle level demonstrate why threshold superimposition is not possible, the decomposition is not increasing.

5.3.4 Gray-scale Limitations

Connected operators extend to gray-scale quite readily [11, 65]. A requirement however, is that for threshold sets at each gray level either the same connectivity class is used, or that the connectivity classes form a connectivity pyramid [57]. For anti-extensive filters this means that the connectivity class used at level h is a subset of that at any level $h' < h$, which guarantees that any connected component of level h is also a connected set at level h' . This property is violated by π -connectivity. An example is shown in the schematic of Fig. 5.2 where given a decomposed gray-scale image f and a connectivity mask m such that $m < f$ the operator handling regions in $T_h(f) \setminus T_h(m)$ extracts connected components which are not nested along the intensity range H . The first image from the left shows a three level gray-scale image followed by its decomposition from h_0 to h_2 (second image). The third shows the corresponding connectivity mask generated by a structural opening with a square

SE. The next two images show the various connected components using a contraction-based m and π connectivity opening respectively. The last case shows clearly that although the squares which are the stable components according to the m -connectivity class are nested appropriately, the bridging regions which are missing from the connectivity mask violate this nesting property and as such

$$\Gamma_x(T_{h_2}(f)) \setminus \Gamma_x(T_{h_2}(m)) \not\subseteq \Gamma_x(T_{h_1}(f)) \setminus \Gamma_x(T_{h_1}(m)) \quad (5.21)$$

for $h_1 < h_2$. The shadowed areas in the middle plane highlight the two regions which generate this nesting conflict. In the next section we show how to use π -connectivity in gray-scale pattern spectra nonetheless.

5.4 Gray-scale Pattern Analysis

5.4.1 Gray-Scale Pattern Spectra Using Max-Trees

Existing methods for computing connected pattern spectra for a gray-scale images require that the corresponding granulometries extend to gray-scale [11, 85, 89]. Under this condition, the gray-scale pattern spectrum is given by replacing the Lebesgue measure with the integral of f (sum of the gray levels) over the image domain. In the discrete case, like with binary images, computing $s_{\gamma_r}(f)$ requires a repetitive filtering by each γ_r , in ascending order of r . At each filter step the sum of gray-levels s_r of the resulting image is computed and the pattern spectrum value at r is given by subtracting s_r from s_{r^-} , with r^- the scale immediately preceding r . In the case of π -connectivity however, the lack of a direct gray-scale extension prevents repetitive filtering. To compute a gray-scale (pseudo) pattern spectrum based on π -connected operators we look into methods that do not require filtering. Urbach et al. used such methods [85] based on connected operators and Max-Trees.

The Max-Tree [65] is a rooted, unidirected tree in which the node hierarchy corresponds to the nesting of peak components given a gray-scale image. A peak component P_h at level h is a connected component of the thresholded image $T_h(f)$ while a flat-zone [66] at level h is a set containing all the pixels of a peak component which are at level h in f . Each tree node C_h^k (k is the node index) contains the sum of the pixels found in all the flat-zones of a given peak component at level h . In addition each node except for the root, points towards its parent $C_{h'}^{k'}$ with $h' < h$. The root node is defined at the minimum level h_{\min} and contains the set of pixels belonging to the background. The algorithm, used primarily for anti-extensive attribute filtering, runs a three-stage process in which the construction of the tree and the computation of node attributes is independent of filtering and image restitution. During the construction stage every pixel visited contributes to the auxiliary data buffer associated to the node it belongs to. Once a node is finalized, its parent inherits these data and re-computes its attribute. Inheritance in the case of attributes such as area/volume is a simple addition

while for more complicated attributes such as the non-compactness measure of (5.6) the accumulation relies on more sophisticated attribute handling functions described in [57, 85].

Computing the pattern spectrum using Max-Trees becomes essentially an accumulation procedure. The method scans the tree structure by visiting all nodes from h_{\min} to h_{\max} and retrieves the attribute measures of the corresponding peak components. Using some binning function (see next subsection) this measure is used to place the corresponding peak component to the appropriate spectral entry. The contribution of each peak component is given by the product of its area with the gray-scale difference from its parent. Each peak component belonging to a given class updates the class *energy counter* by accumulating its product to the existing value. We conclude Urbach's method to the following statement:

$$(s_{\gamma_r}(f))(u) = \sum_{h=h_{\min}}^{h_{\max}} \sum_{\substack{k: C_h^k \neq \emptyset \wedge \\ Bin(P_h^k)=u}} A(P_h^k) \times \Delta h^k \quad (5.22)$$

where Δh^k is the gray-scale difference between the k^{th} node at level h and its parent, and Bin the binning function. If we wish to compute the same sums on a level basis instead of using the Max-Tree dynamics given by the term Δh^k , the same expression reduces to:

$$(s_{\gamma_r}(f))(u) = \sum_{h=h_{\min}}^{h_{\max}} \sum_{k: Bin(P_h^k)=u} A(P_h^k) \quad (5.23)$$

That is, for every level accumulate the area of all peak components whose attributes fall within the bounds of class u . Since π -connected operators are limited to binary sets only, using this formula we can compute a maximum of $H - 1$ Max-Trees, one for each binary image from the threshold decomposition of f . This is for structures in $T_h(f) \setminus T_h(m)$ since the spectrum entries for stable components in $T_h(m)$ are computed using (5.22). For each threshold set at level h (5.23) becomes:

$$(s_{\Gamma_r}(T_h(f)))(u) = \sum_{k: Bin(P_h^k)=u} A(P_h^k). \quad (5.24)$$

5.4.2 Binned 2D Shape-Size Spectra

Multi-dimensional spectra have been used before to sort connected components based on several attribute measures [85]. For the purposes of this work we consider a joint 2D shape-size pattern spectrum that features the non-compactness attribute of (5.6). The method we present for its computation relies on the Max-Tree structure whose corresponding connected components are computed based on the contraction-based π -connectivity opening of (5.3). The procedure is summarized in the following:

Algorithm 1. Computing 2D shape-size binned pattern spectrum s using the Max-Tree for a contraction-based π -connectivity with N_a shape and N_b size classes.

1. *Mask Generation:* Compute the opening transform of the input image for a given SE.
2. *Stable Components:* Compute the Max-Tree of the gray-scale mask image.
3. *Auxiliary data:* As the Max-Tree is built, compute the area $A(P_h^k)$ and the moments of inertia $I(P_h^k)$ corresponding to each peak component P_h^k .
4. *Spectrum Initialization:* Set the $N_a \times N_b$ elements of the spectrum array S to zero.
5. *Spectrum Update:* For each node C_h^k
 - Compute the size class r from the area $A(P_h^k)$.
 - Compute the shape class s from $I(P_h^k)/A^2(P_h^k)$.
 - Compute the gray-level difference Δh^k between the current node and its parent.
 - Add the product of Δh^k and $A(P_h^k)$ to $S[r, s]$.
6. *Non-Stable Components:* For all gray levels threshold both f and m and compute the binary mask-complement image. For each binary image:
 - Compute the binary Max-Tree.
 - Compute the auxiliary data and size/shape classes as above.
 - For each node of the tree update the spectrum as above.

In this algorithm if we were to compute a pattern spectrum based on contraction-based m -connectivity openings, there are certain simplifications which can boost its performance. Since m -connected operators are used to construct gray-scale granulometries the need to threshold f and m would no longer exist. In fact, using the Dual-Input Max-tree algorithm from [57] in step 2 would be sufficient for skipping step 6. In practice however, since singleton sets, just like noise, do not contribute particularly valuable information, it is usually sufficient to compute the Max-Tree of the mask image only. For the transformation of the attribute values into the corresponding bins we use the heuristic function presented in [85]. A class c is given by:

$$c = \left\lfloor \frac{\log_2(v) - \log_2(D_0)}{\log_2(D_1) - \log_2(D_0)} N_c \right\rfloor, \quad (5.25)$$

where $\lfloor \dots \rfloor$ denotes the floor function, v is the attribute value, N_c the number of classes, and D_0 and D_1 the lower and upper bounds of the range of interest of the attribute values.

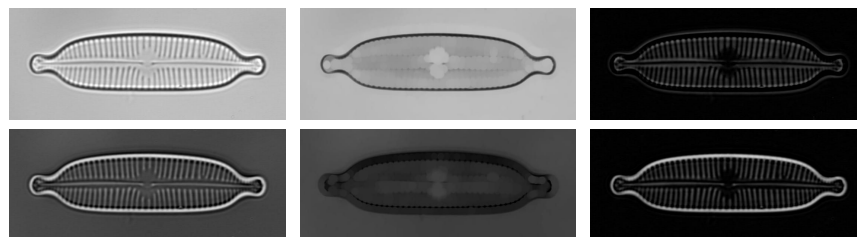


Figure 5.3: Diatom - (left column from top) The original image and the inverted copy; (middle column) the two connectivity masks; (right column) the superimposed unprocessed threshold sets (top hats).

5.5 Diatom Identification Experiments

The experiments described in this section aim at highlighting the significance of structures discarded by contraction-based m -connectivity openings. We chose a diatom image classification problem for this purpose using the π -connected pattern spectra of the image set as feature vectors. Similar experiments were conducted in the past using pattern spectra based on standard connected operators [12]. We follow similar procedures to allow comparisons between the two methods and report on the overall classification performance.

5.5.1 The ADIAC Diatom Image Database

Diatoms are a large and ecologically important group of unicellular or colonial algae which are found in almost all aquatic habitats. Their silica cell walls consist of two halves called *valves* and together with the pattern of pores (internal valve texture also called *ornamentation*) and other valve markings, provide the information needed for species or *taxa* identification.

The experiments that follow make use of two sets of diatom images obtained from the publicly available ADIAC database which can be found at <http://www.ualg.pt/adiac/pubdat/pubdat.html>. The first set referred to as *mixed genera* consists of 781 images representing 37 distinct taxa, and the second, the *Sellaphora pupula*, of 120 images from 6 different subspecies of *Sellaphora*. For both sets each taxa or subspecies is represented by at least 20 images. Moreover, for each of the 8-bit gray-scale images a contour file is given to mask out regions outside the diatom valve. Acquisition and preprocessing methods as well as image features and other details are available in [12].

5.5.2 Experimental Methods and Parametrization

For each of the two image data sets we replicate the experimental procedure followed by Urbach et al. [85] only instead of computing connected pattern spectra based on the standard

connectivity, we experiment with m and π -connectivities. We compute the respective spectra for connectivity openings with 5 different sizes of circular structuring elements starting from radius 3 until 11 incremented each time by 2 pixels. Prior to each run, we compute the bin extrema D_0 and D_1 for each of the two classes as in (5.25). The values are obtained by a scan through the entire image database. The entries of the size class are scaled with the pixel width associated with each image and so are the extrema.

In each experiment we produce a feature vector of 600 elements. The first 300 correspond to processes on the original image and the remaining 300 on the inverted copy. This is done to capture information from both bright and dark patterns in the images. Each set of 300 elements corresponds to a 2D matrix mapped in a lexicographic order to a 1D vector. For each matrix, x-dim. always refers to the non-compactness attribute and y-dim. to the size attribute. The 600-long vector is a concatenation of two such vectors and is complemented by some additional information to meet the classifier's input specifications.

5.5.3 The C4.5 Decision Tree Classifier

To carry out meaningful comparisons we employed the same decision tree classifier built with the C4.5 algorithm as in [12,85]. To compensate for classifier instabilities using a single decision tree, we use *bootstrap aggregation* or *bagging* with the same procedure reported in [85]. Briefly this can be summarized into the following. Firstly, for each image database we divide the total number of images into two subsets, the training and the test set. The latter one contains roughly 25% of the total images per class. To generate the decision tree forest, we select randomly a number of images from the training set which we group into 25 smaller subsets. A single decision tree is built for each set separately, a process which is known as bootstrapping. An accuracy measure described in [85] is then used to evaluate each of the decision trees followed by a majority vote on their outcome (aggregation). We repeat this procedure over 10 times on newly created training sets and obtain the overall classification performance by averaging the individual outputs. An error estimate is computed using cross-validation.

5.5.4 Experiments

The first experiment trains the classifier using π -connected pattern spectra. We run the same experiment with 6 different spectral arrangements. That is, we used three sets of extrema and 2 different spectral layouts, i.e. 15x20 and 20x15. The sets of extrema are the ones provided by Urbach et al., the absolute extrema computed by our scan routine and the extended extrema which are the same as the absolute only excluding small particles by multiplying D_0 of the size class by 3. Table 5.1 lists Urbach's extrema which are given for the *mixed genera* data-set only and the set of absolute extrema that we have computed. Note that in Urbach's experiments D_0 for the size class is also multiplied by 3 to reduce the effects of noise.

Table 5.1: Pattern spectrum extrema for both image data-sets.

SE radius	<i>Mixed Genera</i>			
	Size		Shape	
	D_0	D_1	D_0	D_1
3-9	0.000983	2050.0	1.0	446.68
11	0.000983	2050.0	1.0	395.907
Urbach	0.003	7198	1.0	328.1
<i>Sellaphora pupula</i>				
3-11	0.0025	325.731	1.0	153.963

The highest performance in all scales for both image data-sets was achieved using the extended extrema, i.e, excluding singletons which overflow the first bin of the spectrum. To avoid this in the second experiment where we train the classifier with m -connected pattern spectra, we compute ordinary Max-Trees from the gray-scale connectivity masks associated to each input image. This essentially discards the middle term of (5.3) and resolves the overflow issue. The classification performances for both image data-sets together with the error estimates are listed in tables 5.2 and 5.3. We mark with bold numbers the best result in each case. We observe that the best classification performance is given for a 20×15 spectral layout in both data-sets and for relatively small radii of the structuring elements used.

5.5.5 Performance Optimization Using Combined Methods

The second of the two morphological based methods for diatom feature extraction reported in [12] uses contour analysis by morphological curvature scale spaces [31–33]. We use this in combination with the π -connected pattern spectra presented in this paper and the method of Urbach et al. [85] to optimize the classification performance in the case of the mixed genera data set.

Urbach’s method which processes comparatively larger structures on the diatom valves achieves a classification accuracy of 91.1% in a 15×20 spectral layout. Using the π -connected pattern spectrum alone, we fail to raise this figure further while a small improvement appears (91.46%) if concatenating the two vectors, the original by Urbach and ours, into a 1200-long new one. The increase is limited most probably due to feature correlations which degrade the classifier’s stability (error estimate is 5.49%). To reduce this, we create for each image a 600-long vector made of the average values between the respective members of the 5 feature vectors (one for each scale). We subtract each new member from the corresponding member in Urbach’s feature vector and concatenate the resulting vector with Urbach’s original. Note that in order to compute an average vector, the binning and spectral

Table 5.2: Classification performance for the mixed genera data-set. By comparison: 4-connectivity used by Urbach et al. yields $91.1 \pm 1.6\%$ performance using 15×20 binning.

SE radius	π -connectivity				SE radius	m -connectivity			
	20×15		15×20			20×15		15×20	
	$\mu(\%)$	σ	$\mu(\%)$	σ		$\mu(\%)$	σ	$\mu(\%)$	σ
3	88.47	3.79	89.20	4.47	3	88.37	3.82	88.27	3.63
5	90.70	2.71	87.70	3.10	5	86.86	3.16	86.16	4.88
7	90.64	5.02	90.05	4.52	7	84.75	6.21	85.08	6.68
9	89.89	3.82	89.35	3.10	9	82.38	5.18	84.00	5.21
11	89.45	4.41	89.79	3.47	11	81.02	4.48	82.64	7.62

Table 5.3: Classification performance for the Sellaphora pupula data-set. By comparison: 4-connectivity used by Urbach et al. yields $78.00 \pm 2.14\%$ performance using 15×20 binning.

SE radius	π -connectivity				SE radius	m -connectivity			
	20×15		15×20			20×15		15×20	
	$\mu(\%)$	σ	$\mu(\%)$	σ		$\mu(\%)$	σ	$\mu(\%)$	σ
3	83.00	1.37	76.33	1.97	3	75.00	2.61	71.66	1.91
5	78.66	1.28	75.00	1.80	5	69.66	1.57	70.00	2.68
7	77.00	2.16	79.33	2.03	7	66.33	2.11	69.66	2.07
9	73.66	2.10	71.00	1.34	9	69.66	2.07	66.00	3.34
11	69.33	1.72	75.33	1.85	11	63.33	2.23	69.33	2.13

layout for each scale must be the same. As such we don't use our optimal setup but instead the results obtained using Urbach's spectral extrema for a 15×20 layout. Using this new set of feature vectors (referred to as *combined spectra*) the classifier achieves a prediction accuracy of 92.92% with a slightly reduced error estimate of 3.61%.

A look through the individual connected components associated to different spectral bins reveals that contour structures are poorly represented in both pattern spectrum-based methods. To account for fragmentations and incomplete boundaries we employ the method of Jalba et al. [32,33]. For each image this contour-based method yields a 66-long vector. Concatenating our vector of the combined spectra with Jalba's we raise the classifier's prediction rate to 95.18% with a considerably lower error estimate of 1.92%. Furthermore, if instead of the average vector in the combined spectra we use the best performing vector with Urbach's extrema and in a 15×20 layout and concatenate this with Jalba's vector this figure raises slightly further to 95.78% at the expense of an increased σ of 2.52%. This small increase in the error estimate is expected since in the first case the average vector ensures better stability.

Table 5.4: Classification performances on the mixed genera data set using combinations of feature extraction methods. The term and above indicates concatenation of vectors.

Methods	Clas. Performance		
	$\mu(\%)$	σ	Vector size
π -conn. pattern sp. alone	90.7	2.71	600
std. conn. pattern sp. alone	91.1	1.6	600
std. pattern sp. and π -conn. pattern sp.	91.46	5.49	1200
combined spectra	92.92	3.61	1200
std. pattern sp. and contours	93.94	3.5	666
π -conn. pattern sp. and contours	94.05	3.12	666
combined spectra and contours	95.78	2.52	1266
all methods from [12]	96.9	1.2	329
same with robust features only	95.5	1.5	17

The significance of contour information can also be seen if combined with each of the two spectral-based methods separately. A summary of these results and the best classification rates achieved using combinations of other not-necessary morphological methods is given in Table 5.4.

5.6 Discussion of Results

In the first set of experiments we train the classifier with π -connected pattern spectra. We see that in the case of the mixed genera data-set, we obtain a rather stable performance throughout the five scales and for both types of spectral layouts. This is due to differences in ornamentation between diatom species which result in fragmentations at different scales. Having a limited subset of diatoms affected by this operation at a given scale, the method fails to contribute in improving further the classification accuracy. The success rate of the classifier however compares with that obtained using Urbach’s method.

By contrast, in the case of the *Sellaphora pupula* data-set where different subspecies differ little in ornamentation, we see major variations in classification accuracy as the scale changes. Notably, the larger the radius of the structuring element used the further the drop in success rate. This suggests that for the specific species the fragmentation which occurs at the first scale separates thin elongated features from larger structures providing a more accurate characterization of the ornamentation. This when compared with Urbach’s method run on the entire data-set as opposed to a limited subset reported in [12], yields a gain of 5% in success rate.

Both data-sets show preference in higher bin resolution for the non-compactness attribute. This suggests that the fragmentation of the diatom ornamentation contributes more

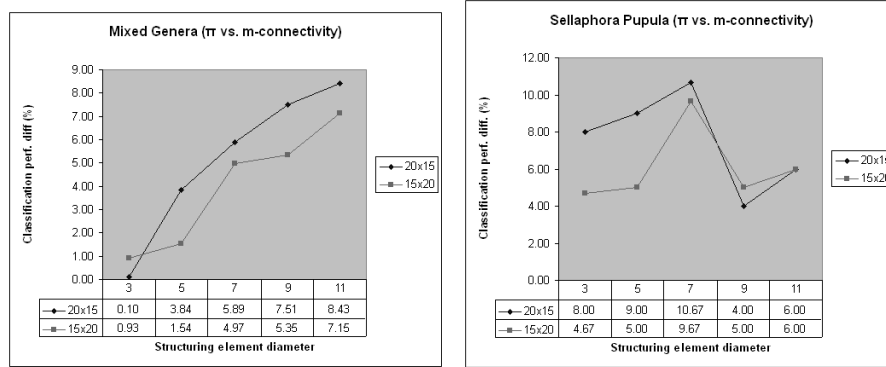


Figure 5.4: Difference in classification performance when training the classifier with π and m -connected pattern spectra. (left).

to shape rather than to size information.

The second set of experiments makes use of m -connected pattern spectra. The observation of classifier instabilities when including large numbers of singletons at the very first bin of the 2-D spectrum led to their exclusion. This essentially reduces to computing the standard connected pattern spectrum on the contracted connectivity masks. In both data-sets we see a progressive decline in classification success as the size of the structuring element increases. This is expected due to the incremental loss of information. It is remarkable though that even for considerably large structuring elements (diameter of 23 pixels) this spectrum based method remains robust and yields a classification success which outperforms many of the other methods reported in [12]. For small values of SE radius much of the noise in the mask which is computed with a structural opening on the original image, is suppressed. As such the classification performance remains high and the feature extraction process essentially becomes equivalent to Urbach's method running on smoothed images.

The graphs in Fig. 5.4 illustrate how the difference between classification success rates using π vs. m -connected spectra changes over scale for each data-set.

As can be seen in Fig. 5.4, in the case of the mixed genera - left graph, the very small difference recorded in the first scale for the 20×15 layout suggests that the influence of ornament fragmentation there is minimal. Since in all methods we discard singletons and small objects with area up to 3 pixels, most of the structures in each $X \setminus M$ must be within this size range. The classification difference which appears as an increasing function of scale is upper bounded by a SE radius value above which each $M = \emptyset$ and consequently $X \setminus M = X$. For the *Sellaphora pupula* set this function is not increasing since for certain SE radii there can be common features between the subspecies that when removed or detached from the remaining ornamentation provide a set of more distinctive descriptors to the classifier. We

see such an example for the SE radius 9 in the right graph of Fig. 5.4. The function however is upper bounded in the same way as with the first case. Concluding on this comparison, we see that the connected pattern spectra method using contracted masks, when based on π -connectivity outperforms the equivalent based on m -connectivity under all types of spectral configurations. This holds for both data sets.

The last set of experiments targets the performance optimization using a combination of the two spectral-based methods together with features obtained by contour analysis using morphological curvature scale spaces [32, 33]. Using the two spectral methods in a way described in the previous section to minimize the feature correlations yields a small performance gain of 1.82%. Note that a number of other methods were tried such as multi-scale and weighted multi-scale sums but none of them succeeded in overcoming Urbach's result of 91.1%. This suggests that texture based information, although considered the best feature descriptors from the comparison in [12], can reach a certain limit in multi-species classification success beyond which further features and of different nature are required. The morphological method of Jalba et al. [33] focuses on contour information instead of the diatom ornamentation and when used separately it reaches a classification success rate of up to 91.3% with $\sigma=5.0\%$. The contour descriptors complement the combined spectrum-based method and as such reduce the error estimate while boosting the overall performance to 95.2%. This is comparable to the best performance reported in [12] by using all 17 methods which were available for this purpose.

5.7 Conclusions

In this paper, starting from Serra's work on image partitions [72] we have presented a new type of connection, the π -connectivity class aided by connectivity masks, which can be used in ways analogous to second-generation connectivity. The steps we use in our proof for establishing the π -connection provide an alternative way to prove the validity of the *mask-based* connectivity [57] and are applicable in establishing other types of connections trivially.

The strength of π -connected operators is in contraction based problems where the handling of pathwise connected regions otherwise treated as singletons, allows the assignment of meaningful attributes and thus further processing. This in part resolves the problem of oversegmentation [54, 92], but due to limitations in extending π -connected operators to gray scale, developing efficient attribute filters remains a topic for further research. The same limitations prevent the introduction of gray-scale granulometries and therefore connected pattern spectra defined in the conventional way. Using the method from [85] we have introduced pseudo pattern spectra and showed that these can be adopted trivially to compute a gray-scale spectrum based on π -connected operators. A brute-force algorithm is also given.

Classification experiments on two diatom image data-sets showed that the use of pattern spectra associated to contraction-based π -connected operators as feature vectors outper-

forms their counterparts associated to contraction-based m -connected operators. Comparisons were also made with spectra associated to standard connected operators. The results in the case of the *Sellaphora pupula* data-set indicate that the fragmentation of ornament structures enhances the differentiation between subspecies of the same family and yields higher classification success rate.

Comparing the classification performance of this method on the genera pupula with other methods reported in literature we achieve a similar rate to the best reported which again uses pattern spectra only based on the standard connectivity. This suggests that the spectral methods alone are limited. Combining the two methods and adding contour descriptors however yields a success rate comparable to the one based on all methods combined (reported in [12]). The obvious advantages in this case is the far smaller number of methods needed to reach this rate and not having the need of manually selecting the best performing features.

In future work we expect to increase these figures further by using different classifiers while resolving further feature correlations that can reduce the size of the feature vectors used. In addition, further work can be done in deriving appropriate filtering rules to extend the π -connected operators to gray-scale directly and thus implement more efficient algorithms for both filtering and pattern spectra.

Chapter 6

Hyperconnected Attribute Filters Based on k -Flat Zones for 3D Medical Imaging

If knowledge can create problems, it is not through ignorance that we can solve them.

Isaac Asimov

Abstract

In this paper we present a new method for implementing attribute filters, involving contrast information together with structural characteristics. The filters, instead of the standard 4 and 8 connectivity, rely on the recently introduced notion of hyperconnectivity. The theory of hyperconnections, just like with ordinary connections, is given by means of classes and provides an axiomatic definition of set overlap. The filters we propose work on hyperconnected sets of maximal extent that are derived from a “base” connectivity class. This allows us to use standard image representation algorithms like the Max-Tree for their efficient computation. The method is implemented in the form of a filtering rule suitable for handling both increasing and non-increasing attributes. In our experiments we show that fine details, usually observed at higher levels, that normally would fail the filter’s criteria, are preserved if found within a certain contrast range from the objects of interest. On the contrary, undesired structures resting on the background that would previously be accepted by the filter are now suppressed. We demonstrate the usability of this new framework, on non-increasing shape filters operated on 3D medical data sets and we compare the results with those of the same filters configured with standard connectivity. Our findings show an increased robustness to noise while maintaining the advantages of previous methods.

6.1 Introduction

MORPHOLOGICAL operators [26, 67] in modern image analysis are a set of powerful, robust and computationally efficient tools that find use among other areas, in image filtering [56, 66, 88, 94], segmentation [16, 23, 35], and visualization [46, 90]. They extend to gray-scale trivially and depending on whether they operate on points or sets of pathwise

connected points, i.e. *connected regions*, they are referred to as structural or connected operators respectively.

Filters based on connected operators [27, 29] can either remove a connected component or retain it unmodified but cannot introduce new ones. This is an edge preserving property highly desirable in many applications. If this decision is based on some attribute measure such as area or elongation, they are referred to as attribute filters [11]. Like all connected operators, attribute filters rely on some notion of connectivity, commonly the 4 and 8 pixel adjacency relations [37]. This graph-based definition is part of a wider lattice-theoretic connectivity framework that associates directly to connected operators [45, 68, 69]. This is through the mathematical construct of *connectivity classes* which can also be used to introduce several generalizations such as the second generation [7, 57, 62] and partition induced π -connectivity [59, 70]. In both cases the objective is to set criteria for constraining or enriching the ways images are connected. This can yield clusters of objects or separate touching regions into several components.

Attribute filters based on standard connectivity or its derivatives are in general insensitive to contrast information. The flat treatment of gray-scale images often requires a high enough attribute threshold to remove noise or other particles such that a number of fine but of low attribute measure details, found at the higher levels of the objects of interest, are lost.

Second generation connectivity can in part resolve this by treating same level components, close enough to each other as clusters of connected components. Depending on the attribute considered and on the position of the components with respect to each other this can yield a sufficiently high attribute measure to satisfy the filter's criterion. A problem with this approach is that noise gets clustered too, and sometimes together with the objects of interest. Methods to reduce this involve different ways of computing the cluster attributes [56] and the use of orientation information instead of just distance measures between objects to define the clusters [57]. Though efficient in some cases, in 3-D it does cause significant computational overhead.

Salembier et al. [65] in their work on anti-extensive attribute filters presented a contrast sensitive method involving a "soft" binarization of gray-scale images. In this case, λ -flat zones are considered in which from any given pixel in the component, any other pixel of the same component can be reached through a path, in which neighbours differ by no more than λ . Though this prevents the oversegmentation issues of strict flat-zones, it aggravates the leakage problem common to all connected filters [65]. This effect can clearly be seen in Fig. 6.1, where even at $\lambda = 1$ the entire image is just one λ -flat zone, because a path from any pixel to any other can be made in which grey levels between adjacent pixels is no more than 1. Many attempts at resolving this have been proposed (for a review see [76]), but few if any have been used in attribute filtering.

In this paper we propose a new method for contrast sensitive attribute filters based on hyperconnectivity [9, 69]. Hyperconnectivity is employed for clustering connected regions

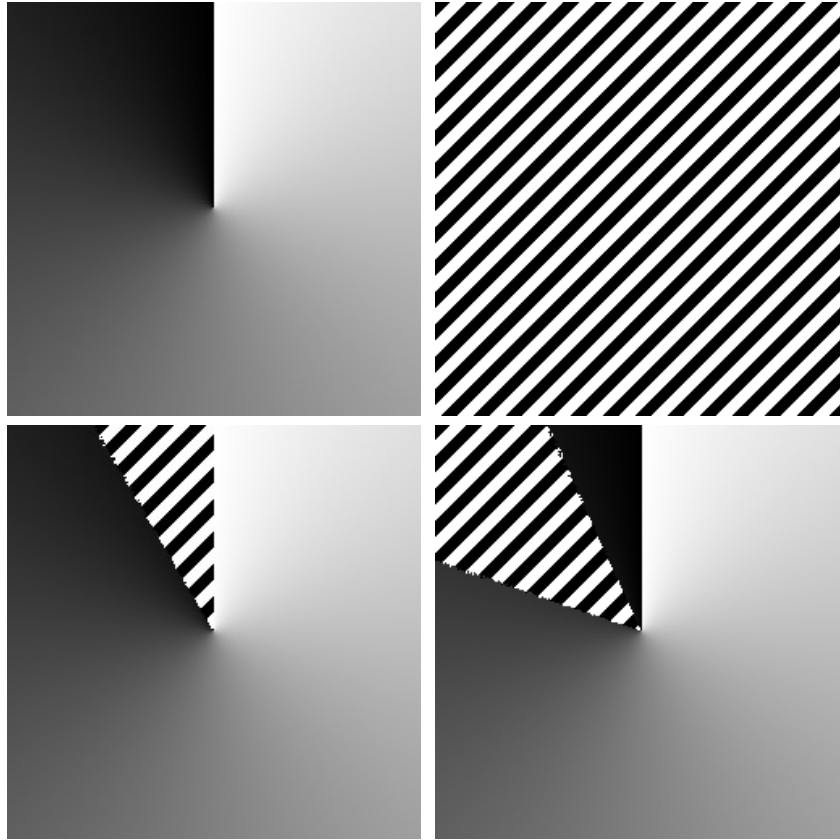


Figure 6.1: The difference between k -flat zones and λ -flat zones: (top left) image showing circular gradient; (top-right) λ -flat zone (indicated as hatched area) for $\lambda = 1$; (bottom row) two k -flat zones for $k = 16$. Note how the two k -flat zones overlap.

along the intensity range and from different threshold sets rather than nearby regions as in clustering-based second generation connectivity.

In this case we will work with k -flat zones, which are defined as connected regions of maximal extent, in which the total grey level variation is no more than k . This restriction to grey-level range automatically restricts the size to which the regions can grow, as can be seen in Fig. 6.1. This does yield overlapping pseudo-flat zones, and thus a cover of the image domain, but we will show that this does not prevent the definition of attribute filters.

A cluster in the proposed scheme is characterized by its reference level h and its depth k . The depth, which is a global parameter, specifies an intensity range above h . All nested

connected components in that range are registered (not exclusively) as members of the cluster defined at h . A filter's decision on a cluster propagates to all its members but clusters may also overlap by sharing their members. This means that though a cluster may fail a filter's criterion certain members might be preserved if they also belong on a surviving cluster or the other way around, i.e. components that satisfy the criterion will be rejected if they don't belong in a surviving cluster. Thus we judge components not only based on their attribute measure but also based on what they rest on.

In Section 6.2 we briefly present some connectivity concepts and discuss on attribute filters and ways for extending them to gray-scale. In Section 6.3.2 we start off by giving a short introduction on *covers* and hyperconnectivity and we present the proposed method. An algorithm together with an implementation analysis for a suitably adopted filtering rule are given in Section 6.4. Experiments on 3D medical data-sets together with a short discussion and conclusions are given in sections 6.5 and 6.6 respectively.

6.2 Connections, Partitions and Operators

6.2.1 Connections and Partitions

The concept of connectivity in discrete image analysis provides the means to group pixels into structures with specific topological properties. In mathematical morphology a common way of addressing connectivity is through the set-oriented definition of *connectivity classes* or *connections* [45, 68].

Definition 18. *Let E be an arbitrary non-empty set. A connectivity class or connection \mathcal{C} on E is any family in $\mathcal{P}(E)$ that satisfies:*

1. $\emptyset \in \mathcal{C}$ and for all $x \in E$, $\{x\} \in \mathcal{C}$,
2. for any $\{A_i\} \subseteq \mathcal{C}$ for which $\bigcap A_i \neq \emptyset \Rightarrow \bigcup A_i \in \mathcal{C}$

This means that both the empty set and singleton sets, denoted as $\{x\}$, are connected, and any union of elements of \mathcal{C} which have a non-empty intersection is also connected. The members of \mathcal{C} are called *connected sets* and are element groupings of E .

Connected sets that share a common point x in their intersection can be addressed as a single connected entity by computing their union according to Def. 18. This is known as a *connected component* or *grain* C_x of X and is a set of maximal extent, i.e. given a set C_x there can be no other set $C'_x \supset C_x$ such that $C'_x \subseteq X$ and $C'_x \in \mathcal{C}$. Given a point $x \in E$, the connected component C_x of a set X can be extracted by a *connectivity opening* which is an operator defined as:

$$\Gamma_x(X) = \bigcup \{A_i \in \mathcal{C} \mid x \in A_i, A_i \subseteq X\} \quad (6.1)$$

for every $X \subseteq E$. The connectivity openings are algebraic openings and the datum of a connectivity class C in $\mathcal{P}(E)$ is equivalent to the family $\{\Gamma_x, x \in E\}$ [62, 68, 69].

The notion of connectivity as given by Def. 18 is referred to as standard connectivity and an example is the 4 and 8 graph-based adjacency relations. The associated connectivity openings safeguard the topological properties of connected components and essentially reject any point which is not path-wise connected to x .

Connected components form a partition of the image domain. Partitions like connections describe element groupings on E . The formal definition as given in [70] is the following:

Definition 19. *Let E be an arbitrary set. A partition \mathbf{P} of E is a mapping $x \rightarrow \mathbf{P}(x)$ from E into $\mathcal{P}(E)$ such that*

1. *for all $x \in E : x \in \mathbf{P}(x)$,*
2. *for all $x, y \in E : \mathbf{P}(x) = \mathbf{P}(y)$ or $\mathbf{P}(x) \cap \mathbf{P}(y) = \emptyset$.*

$\mathbf{P}(x)$ is called the class of the partition of origin x . The two conditions indicate that classes $\mathbf{P}(x)$ occupy the whole space E and that two distinct classes have no common point.

Partition classes as opposed to connected components, do not necessary contain elements from the foreground sets only. Because of this, establishing a relation with a connection requires the use of connectivity openings which naturally separate background from foreground components [68, 72].

Definition 20. *Given a partition \mathbf{P} of the space E , all the subsets of each class $\mathbf{P}(x), x \in E$, of the partition generate a family conditionally closed under union given by*

$$\mathcal{C}^\pi = \{A \cap \mathbf{P}(x), x \in E \text{ and } A \in \mathcal{P}(E)\}. \quad (6.2)$$

We call \mathcal{C}^π a *partition-induced* (pi) or π -connection [59].

6.2.2 Attribute Filters

In mathematical morphology an operator is called a filter if it is increasing and idempotent [27, 28, 68]. For any two sets $X, Y \subseteq E$, increasingness implies that if $X \subseteq Y \Rightarrow \Psi(X) \subseteq \Psi(Y)$ and idempotence that $\Psi(\Psi(X)) = \Psi(X)$. In the case of connected operators, a filter $\Psi : \mathcal{P}(E) \rightarrow \mathcal{P}(E)$ interacts with connected components rather than individual pixels. In the more specific class of attribute filters, connected components are preserved unmodified if they meet some pre-specified attribute criterion Λ or removed otherwise.

Attribute criteria for connected components $C_x \subseteq X$ are typically given in the form of:

$$\Lambda(C_x) = Attr(C_x) \geq \lambda \quad (6.3)$$

with $Attr(C_x)$ some real-value attribute of C_x , and λ an attribute threshold. They are put in place by means of a trivial opening which is an operator $\Gamma_\Lambda : \mathcal{C} \rightarrow \mathcal{C}$. For a connected component C_x :

$$\Gamma_\Lambda(C_x) = \begin{cases} C_x & \text{if } \Lambda(C_x) \text{ is true} \\ \emptyset & \text{otherwise.} \end{cases} \quad (6.4a)$$

$$(6.4b)$$

Furthermore, $\Gamma_\Lambda(\emptyset) = \emptyset$.

The attribute filter Ψ^Λ of a set X given a criterion Λ is given by:

$$\Psi^\Lambda(X) = \bigcup_{x \in X} \Gamma_\Lambda(\Gamma_x(X)). \quad (6.5)$$

Let \mathcal{C}_X^* denote the set of all connected components of X . We can then rewrite (6.5) as

$$\Psi^\Lambda(X) = \bigcup_{C_i \in \mathcal{C}_X^*} \Gamma_\Lambda(C_i). \quad (6.6)$$

Depending on whether the criterion is increasing or not, Ψ^Λ is referred to as an attribute opening or thinning respectively (in the anti-extensive case). A commonly used increasing criterion is the area of C_i [14, 87]. Non-increasing criteria are discussed in [11, 29]. In this work we experiment with 3D shape filters that use non-increasing criteria, notably the non-compactness measure of [86, 94]. A 3D structure is non-compact if it is characterized by a high trace of the moment of inertia tensor $I(C)$ compared to its volume $V(C)$. In 3D, $I(C)$ has a minimum for a sphere and increases rapidly as the object becomes more elongated. It is defined as:

$$I(C) = \frac{V(C)}{4} + \sum_{\mathbf{x} \in \mathbf{C}} (\mathbf{x} - \bar{\mathbf{x}})^2 \quad (6.7)$$

and scales with size to the fifth power whereas the volume scales to the third power. Therefore the ratio

$$Attr(C) = \frac{I(C)}{V^{5/3}(C)} \quad (6.8)$$

is a purely shape dependent attribute which can be used to define a filter sensitive to elongated structures.

6.2.3 Extensions to Gray-Scale

Increasing connected filters extend to gray-scale trivially by threshold superposition [43]. Given a gray-scale image $f : E \rightarrow \mathbb{R}$, thresholding f in an increasing order from $h_{\min} + 1$ to h_{\max} yields a stack of nested binary sets. Each binary image at level h is given by:

$$T_h(f) = \{x \in E \mid f(x) \geq h\}, \quad (6.9)$$

and for any two levels such that $h < h'$:

$$T_{h'}(f) \subseteq T_h(f). \quad (6.10)$$

Given a threshold decomposition of f , the response of the gray-scale counterpart of a binary increasing filter Ψ^Λ on each point x of f is given by:

$$\psi^\Lambda(f)(x) = \sup\{h \mid x \in \Psi_\Lambda(\Gamma_x(T_h(f)))\}. \quad (6.11)$$

Thus, the operator ψ^Λ assigns to each x the highest threshold at which it still belongs to a connected foreground component of attribute measure equal or larger than λ . Attribute filters are implemented efficiently on image representation structures with the aid of *filtering rules*. Depending on the rule, non-increasing attributes can also be used to define gray-scale non-increasing grain filters [29], which are idempotent, but not increasing. More on rules and strategies are discussed in Section 6.3.2 and in [65, 85].

Next, we define three types of components used in gray-scale image analysis in relation to connected components for the purposes of threshold superposition. Given a gray-scale image f , a peak component P_h is a connected component of the threshold set at level h [65, 66], i.e.

$$P_h = \Gamma_x(\{x \in E \mid f(x) \geq h\}) \quad (6.12)$$

and a flat zone F_h is a connected component of the set of pixels with level strictly equal to h [66], i.e.

$$F_h = \Gamma_x(\{x \in E \mid f(x) = h\}) \quad (6.13)$$

If a peak component P_h at level h has no neighbors of intensity greater than h , it is called a *regional maximum*.

6.3 Hyperconnections

6.3.1 Hyperconnectivity Classes and Covers

Hyperconnectivity [9, 69] extends the notion of standard connectivity by relaxing the second condition of Def. 18. Instead of using a strict non-empty intersection of sets for their union to be connected, the definition of hyperconnectivity classes involves a *degree* of overlap specified by an *overlap criterion*.

Definition 21. An overlap criterion in $\mathcal{P}(E)$ is a mapping $\perp : \mathcal{P}(\mathcal{P}(E)) \rightarrow \{0, 1\}$ such that \perp is decreasing, i.e., for any $A, B \subseteq \mathcal{P}(E)$

$$A \subseteq B \quad \Rightarrow \quad \perp(B) \leq \perp(A), \quad (6.14)$$

This condition makes it explicit that a non-overlapping family cannot possibly become overlapping by adding more sets. Any $A \subseteq \mathcal{P}(E)$ for which $\perp(A) = 1$ is said to be *overlapping*, otherwise A is non-overlapping. A *hyperconnectivity class* can now be defined as follows.

Definition 22. A hyperconnectivity class $\mathcal{H} \subseteq \mathcal{P}(E)$ with an overlap criterion \perp is a set of sets with the following properties:

1. $\emptyset \in \mathcal{H}$ and for all $x \in E$, $\{x\} \in \mathcal{H}$,
2. for any $\{A_i\} \subseteq \mathcal{H}$ for which $\perp(\{A_i\}) = 1 \Rightarrow \bigcup A_i \in \mathcal{H}$.

It can be seen that all connectivity classes are special cases of hyperconnectivity, in which the overlap criterion is given by:

$$\perp(\{A_i\}) = \begin{cases} 1 & \text{if } \bigcap A_i \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \quad (6.15a)$$

$$(6.15b)$$

Examples are given in [9].

Sets which are members of a hyperconnectivity class are called hyperconnected. In a complete analogy to standard connectivity, we can define hyperconnected components as sets in \mathcal{H} of maximal extent. Given a binary image $X \in \mathcal{H}$ let

$$\mathcal{H}_X = \{A \in \mathcal{H} \mid A \subseteq X\}, \quad (6.16)$$

be the family of all hyperconnected subsets of X . A hyperconnected component of X is a set $C^{\mathcal{H}} \in \mathcal{H}_X$ given by:

$$\mathcal{H}_X^* = \{A \in \mathcal{H}_X \mid \nexists B \in \mathcal{H}_X : A \subset B\}, \quad (6.17)$$

which just states that any hyperconnected component of X has maximal extent, because there exist no hyperconnected subsets of X larger than any of the members of \mathcal{H}_X^* .

Hyperconnected as opposed to ordinary connected components do not necessarily form a partition on X . That is because for any two hyperconnected components $H_i, H_j \in \mathcal{H}_X^*$ which overlap in the sense of *connectivity* (having a non-zero intersection) their union needs not to be a member \mathcal{H} , because $\perp(\{H_i, H_j\}) = 0$. They form a *cover* \mathbf{K} of E instead, which is defined like a partition only dropping the second condition of Def. 19. For the classes of a cover of E we have that $\bigcup_{x \in E} \mathbf{K}(x) = E$, i.e. a partition is a special case of a cover.

Proposition 4. Any cover \mathbf{K} of E with classes $\mathbf{K}(x), x \in E$ induces a hyperconnectivity class $\mathcal{H}^{\mathbf{K}}$ given by

$$\mathcal{H}^{\mathbf{K}} = \{\emptyset\} \cup \{A \in \mathcal{P}(E) \mid \exists x \in E : A \subseteq \mathbf{K}(x)\}, \quad (6.18)$$

associated to the overlap criterion:

$$\perp_{\mathbf{K}}(\{A_i\}) = \begin{cases} 1 & \text{if } \exists x \in E : \cup A_i \subseteq \mathbf{K}(x) \\ 0 & \text{otherwise} \end{cases} \quad (6.19a)$$

$$(6.19b)$$

Proof. The empty set is hyperconnected by the definition. For any of the sets A_i with i from some index set, independent of whether $A_i = \{x\}$ or not, if $\perp_{\mathbf{K}}(\{A_i\}) = 1$ and since $A_i \in \mathbf{K}(x)$, their union yields $\cup A_i \in \mathbf{K}(x) \Rightarrow \cup A_i \in \mathcal{H}^{\mathbf{K}}$.

We can now define hyperconnected attribute filters $\Psi_{\mathcal{H}}^{\Lambda}$ simply by replacing \mathcal{C}_X^* in (6.6) by \mathcal{H}_X^* , i.e.

$$\Psi_{\mathcal{H}}^{\Lambda}(X) = \bigcup_{H_i \in \mathcal{H}_X^*} \Gamma_{\Lambda}(H_i). \quad (6.20)$$

Obviously, $\Psi_{\mathcal{H}}^{\Lambda}$ is anti-extensive, because it can only remove hyperconnected components, not add any. We can simply prove that such a filter is idempotent, by considering that the result of $\Psi_{\mathcal{H}}^{\Lambda}(X)$ is the union of those hyperconnected components of X which meet criterion Λ . Furthermore, all these preserved hyperconnected components are hyperconnected components of $\Psi_{\mathcal{H}}^{\Lambda}(X)$, because $\Psi_{\mathcal{H}}^{\Lambda}(X)$ cannot contain any hyperconnected supersets of hyperconnected components of X , by its anti-extensiveness. Thus, applying $\Psi_{\mathcal{H}}^{\Lambda}$ to $\Psi_{\mathcal{H}}^{\Lambda}(X)$ will only consider those connected components of X which already meet the criterion Λ . Therefore, none are removed, and

$$\Psi_{\mathcal{H}}^{\Lambda}(\Psi_{\mathcal{H}}^{\Lambda}(X)) = \Psi_{\mathcal{H}}^{\Lambda}(X). \quad (6.21)$$

If Λ is increasing, then so is $\Psi_{\mathcal{H}}^{\Lambda}$, as can be seen by considering the hyperconnected components of any two images X and Y with $X \subseteq Y$. For any hyperconnected component $H_i \in \mathcal{H}_X^*$, we have $H_i \in \mathcal{H}_Y$, because although H_i must be a hyperconnected subset of Y , it need not be a hyperconnected component of Y . If $H_i \notin \mathcal{H}_Y^*$, there exists some hyperconnected component of Y which is a superset of H_i . Therefore, for any $H_i \in \mathcal{H}_X^*$ we have

$$H_i \subseteq \Psi_{\mathcal{H}}^{\Lambda}(X) \Rightarrow H_i \subseteq \Psi_{\mathcal{H}}^{\Lambda}(Y), \quad (6.22)$$

because either $H_i \in \mathcal{H}_Y^*$, in which case it will be preserved if it was preserved by Λ , or there is some $H_j \in \mathcal{H}_Y^*$, with $H_i \subset H_j$. In the latter case $\Lambda(H_i) \Rightarrow \Lambda(H_j)$ through increasingness of Λ .

Thus, Hyperconnected attribute filters preserve the main properties of connected attribute filters. However, for a more complete theory we will need to define the axiomatics of the families of operators which extract hyperconnected components, in analogy to families of connectivity openings. Braga-Neto et al. [9] define a hyperconnectivity opening as an operator that given a point x , extracts the union of all hyperconnected sets. It is shown though that the result is not always return a hyperconnected set itself. Defining an operator that yields a hyperconnected set of maximal extent for all $x \in E$ remains an open problem and

it is not dealt with in this paper. For the purposes of our work we employ a specific type of hyperconnectivity for which this analysis is not necessary.

6.3.2 Covers of k -Flat Zones

In [73] Serra et al. showed that the set of flat zones of a gray scale image f constitutes a partition of the space. It can be seen from Def. 20 that this induces a connection \mathcal{C} made up from all the subsets of $\mathcal{P}(E)$ intersected with the classes of the partition, i.e. the flat zones themselves. Using this fact, Salembier et al. [66] presented a rather simpler way of extending attribute filters to gray-scale as follows:

Definition 23. *An operator Ψ acting on gray-level images is said to be connected if, for any f , the partition of flat zones of $\Psi(f)$ is coarser than the partition of flat zones of f .*

In their work on contrast-based connected operators, Serra and Salembier et al. [65, 66] suggested a "fuzzy" equivalent of flat zones, which we will call λ -flat zones. In this case a connected path of pixels exists between any two members of the same λ -flat zones, in which neighbouring pixels differ by no more than λ in grey level. Connected operators extending to gray-scale that work on such components were introduced but as pointed out in [65] they lack idempotence and thus cannot be used to define attribute filters.

Consider now the following entity; a k -flat zone at level h is a set of all path-wise connected pixels with intensities from $h - k$ up to h , i.e. at any point $x \in E$ we have:

$$F_{h,k}(x) = \Gamma_x(\{y \in E \mid h - k \leq f(y) \leq h\}) \quad (6.23)$$

It is obvious that for $k > 0$, the k -flat zones found at all levels $h \in [h_{\min} + k, h_{\max}]$ show overlap and thus do not form a partition of E . Consequently they do not represent a connectivity on E in any way. They do however form a cover of E which from Prop. 4 induces a hyperconnectivity class in the form of (6.18).

Proposition 5. *Let f be a gray-scale image, decomposed to a set of k -flat zones $F_{h,k}$ with $k \in \mathbb{Z}$. The set of all $F_{h,k}$ such that $h \in [h_{\min} + k, h_{\max}]$ induces a hyperconnectivity class given by:*

$$\mathcal{H}^k = \{\emptyset\} \cup \{A \in \mathcal{C} \mid \|f(p) - f(q)\| \leq k \quad \forall p, q \in A\}, \quad (6.24)$$

that is associated with the overlap criterion:

$$\perp_k(\{H_i\}) = \begin{cases} 1 & \text{if } \bigcap A_i \neq \emptyset \wedge \\ & \max_{p,q \in \bigcup A_i} \|f(p) - f(q)\| \leq k \\ 0 & \text{otherwise.} \end{cases} \quad (6.25)$$

Proof. In this proposition we assume a "base" connectivity class that applies to all threshold sets. As such the hyperconnectivity class \mathcal{H}^k is made up of overlapping connected sets thus

$$\mathcal{H}^k \subseteq \mathcal{C}. \quad (6.26)$$

Moreover, from (6.23) and (6.24) it is obvious that any $F_{h,k} \in \mathcal{H}^k$.

Each connected set marked by x is included in a class of a partition according to Definition 19. Due to overlap though, in the case of k -flat zones this yields a cover \mathbf{K} rather than a partition, for which each member of \mathcal{H}^k is a subset of a class $\mathbf{K}(x)$. Since sets $A \in \mathcal{H}^k$ are both connected and hyperconnected we need to ensure that $\bigcup A_i \in \mathcal{H}^k \Rightarrow \bigcup A_i \in \mathcal{C}$. This is provided by the definition of the overlap criterion from which

$$\bigcap A_i \neq \emptyset \Rightarrow \perp_k(\{A_i\}) = 1 \Rightarrow \bigcup A_i \in \mathcal{H}^k, \quad (6.27)$$

and from (6.26): $\bigcup A_i \in \mathcal{C}$. The second term in the conditions giving $\perp_k(\{A_i\}) = 1$ ensures that any hyperconnected set is included in a k -flat zone. To verify that \perp_k is decreasing we look at both terms separately. It is obvious that adding more sets to $\bigcap A_i$ cannot result in \emptyset and for any pair of points in $\bigcup A_i$ there cannot be an intensity higher than k . The inf of these two terms is decreasing.

If k is set to 0 which means that overlap ceases to take place, the k -flat zones become the ordinary flat zones of f defining a partition and thus the expression for \mathcal{H}^k yields the standard connection \mathcal{C} .

Let us now consider the k -flat-zone equivalent of regional maxima. A regional maximum M_h at level h is a 0-flat zone which grey-level h which has neighbours of strictly lower grey level. A k -regional maximum $M_{h,k}$ is a k -flat-zone of level h which has neighbours of grey level strictly smaller than $h - k$. Obviously,

$$M_h = M_{h,0} \subseteq M_{h,k} \quad \forall k > 0. \quad (6.28)$$

These same regional maxima correspond to k -peak components $P_{h,k}$ at their respective grey levels. This means these k -peak components correspond exactly to regular peak components at level $h - k$, i.e.,

$$M_{h,k} = P_{h,k} = P_{h-k}. \quad (6.29)$$

If we extend this definition of k -peak components from only the regional maxima to all peak components we can define a Max-Tree based on k -peak components. If $k = 0$ we end up with the regular Max-Tree (as it should be), but as k is increased we will change the topology of the tree. In particular, we will cluster multiple peak-components P_h^i into their supersets $P_{h,k}^j = P_{h-k}^j$, with i and j some indices. Such clustering can be performed by the dual-input Max-Tree algorithm intended for second generation connectivities [57]. We can achieve this using a mask m defined as

$$m(x) = f(x) + h \quad \forall x \in E. \quad (6.30)$$

Though this approach would work, it is wasteful in terms of memory. Instead, we will introduce a new filtering rule for the regular Max-Tree algorithm, as explained in the next section.

6.3.3 Attribute Filters Based on k -Flat Zones

Attribute filters for gray-scale images work on level components. Filters that rely on standard connectivity have the criterion Λ applied on each peak component and proceed according to filtering rule chosen. The attribute measure of a peak component P_h is given by accumulating the auxiliary data of all its descending components with those of its flat zones. Note that in practice there exist more efficient schemes and an example is the Max-Tree algorithm discussed in the next section. For the purposes of this analysis however we assume this simple approach.

The filter that we propose works on k -peak components instead. The objective is to capture contrast information by controlling the parameter k . That is, to preserve fine details of low attribute measure, usually observed at the higher levels on the objects of interest, while suppress all low contrast structures, like noise patterns which rest on the background, even if they are of high attribute measure.

We have devised two strategies for this purpose. The first involves a tree-based algorithm that encodes the notion of k -peak components in its structure by specifying k in advance. This is based on the dual-input Max-Tree [57] where the mask images provided, are replicas of the original shifted in intensity by k levels up. The method yields clusters of regular peak components with their attributes computed based on the mask image as in [56]. The drawback in this, is that for different values of k we need to recompute the tree structure before filtering.

The second strategy presented here, involves a regular Max-Tree structure where k is set interactively for filtering purposes only, allowing the same rapid visualization as in [90]. The hyperconnected attribute filters introduced are configured with the *subtractive filtering rule* which has been selected for its clear advantages [85] over other rules in handling non-increasing criteria. The subtractive rule is described as a non-pruning filtering strategy. Briefly, in the case of standard connectivity, if a peak component does not meet the criterion Λ its flat zones are lowered in grey level to meet the highest surviving ancestor. The feature differentiating it from the other non-pruning rule, the *direct*, is that it also lowers the intensity of its descendants by the same amount.

Consider now a threshold decomposition of a gray-scale image f . The component $P_{h,k}$ is preserved if $Attr(P_{h-k}) \geq \lambda$ and this decision propagates to all regular peak components up to P_h , independent of whether each one of them separately satisfies Λ or not. If the descendants of any regular peak component P_h have a furthest descendent $P_{h'}$ with $h' < h + k$, this means there is no k -peak component at h in this branch of the tree. The decision

as to whether to reject them or not is no longer based on $\Lambda(P_h)$ but on the attribute value of the ancestor at $h' - k$.

Therefore, unlike classical Max-Tree filtering rules, the k -subtractive rule relies on both checking peak components using criterion Λ and an upward propagation of preserve decisions, within some propagation range k' of the preserved ancestors. The propagation range k' of the preserve status is updated for every new peak component found along the same root-path. For any node preserved because Λ is met, its k' is set to k . If $P_{h'}$ fails the criterion but is still within k' levels from its immediate ancestor P_h which was preserved, it is also preserved but only propagates the preserve status to $h + k' - h'$ levels up from h' . If however, $P_{h'}$ fails Λ and its immediate ancestor is more than k' levels below h' then $P_{h'}$ is removed (and propagation range is set to 0).

We summarize this set of rules to what we call the k - *subtractive* filtering rule which is defined as follows. Let χ denote the characteristic function for a binary image X :

$$(\chi(X))(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise.} \end{cases} \quad (6.31a)$$

$$(6.31b)$$

Definition 24. A gray-scale attribute filter ψ_k^Λ configured with k -subtractive filtering rule is given by:

$$(\psi_k^\Lambda(f))(x) = \sum_{h=h_{\min}}^{h_{\max}} \sum_{i \in I_h^f} \chi(\Gamma_\Lambda^k(P_h^i)), \quad (6.32)$$

in I_h^f is an index set for peak components in image f at level h which Γ_Λ^k is defined as

$$\Gamma_\Lambda^k(P_h^i) = \begin{cases} P_h^i & \text{if } (\Gamma_\Lambda(P_h^i) \wedge \exists P_{h+k}^j \subseteq P_h^i) \\ & \vee \exists P_{h'}^j : (h' \geq h - k \\ & \wedge P_h^i \subseteq P_{h'}^j \wedge \Gamma_\Lambda(P_{h'}^j)) \\ \emptyset & \text{otherwise.} \end{cases} \quad (6.33a)$$

$$(6.33b)$$

Next, we demonstrate in Fig. 6.2 how the k -subtractive filtering rule operates on a 1D signal. Assume that we use a non-increasing criterion and that $k = 3$. In the original signal there exist 4 regional maxima for which we assume that the first three from the left, i.e. P_8^0 , P_7^0 and P_6^1 , fail Λ , while $Attr(P_2^1) > \lambda$. Also, all the other peak components except for the root satisfy Λ . An attribute filter relying on standard connectivity and configured with the subtractive rule would remove the first three regional maxima by lowering their intensities to those of their respective parents and leave the rest of the signal unaffected. In the case in which the same filter is configured with the k -subtractive filtering rule instead, the results vary significantly. Starting with the background we see that since it fails Λ and there is no other node below it, it is rejected and with its propagation range set to 0. The left lobe of the signal has two k -peak components before it gets split in two. Both $P_{4,3}^0$ and $P_{5,3}^0$ are

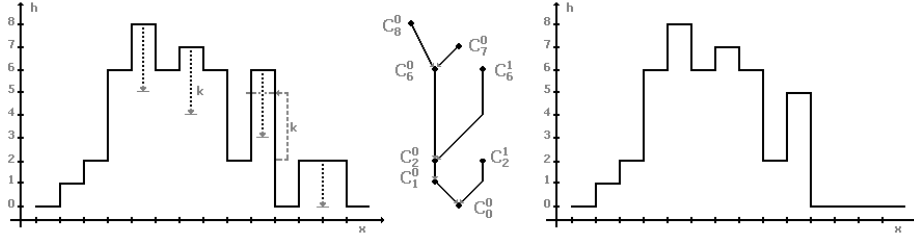


Figure 6.2: The k -subtractive filtering rule for a 1D signal. (from left) The original 1D signal (solid line) with the 4 regional maxima; the Max-Tree of the original signal; the filtered output using the k -subtractive filtering rule.

preserved because they have sufficient contrast and P_1^0 and P_2^0 respectively satisfy Λ . P_2^0 propagates a preserve status for 3 levels up and though this does not affect much in the left group of descendants, it preserves part of the right group (gray dotted arrow) in which all descending peak components fail Λ . That is, the previously rejected regional maximum P_6^1 is only lowered by 1. On the left group again, we keep on finding peak components that satisfy Λ until level $h = 5$. From $h = 6$ and up the contrast range does not permit for any more and the decision on the remaining regular peak components is based on the youngest surviving k -peak component, i.e. $P_{8,3}^0 = P_5^0$. Since they are all within the contrast range of $P_{8,3}^0$ (black dotted arrow), they are preserved as they are. Coming back to the background component, we see that the contrast range of the right lobe is below k , thus there cannot be any k -peak components. The decision of the regional maximum P_2^1 is thus left on the background component which was rejected and with its propagation range set to 0, i.e. P_2^1 though satisfies Λ , is rejected.

6.4 The k -Subtractive Filtering Rule for the Max-Tree Algorithm

6.4.1 The Max-Tree Algorithm

Attribute filters have been implemented efficiently on tree-based algorithms for gray-scale image representation [10, 35, 65]. An example is the Max-Tree introduced by Salembier [65] in the context of anti-extensive attribute filtering. The Max-Tree is a versatile algorithm running a three-stage process in which the construction of the tree and the computation of node attributes is independent of filtering and image restitution. Given a gray-scale image f , the tree structure reflects the nesting order of its threshold sets. The nodes C_h^i , addressed by their level h and index i , correspond to sets of flat zones for which there exists a unique

mapping to peak components:

$$C_h^i = \{x \in P_h \mid f(x) = h\}. \quad (6.34)$$

The tree is rooted and unidirected with its leaves corresponding to regional maxima. The root node is defined at the minimum level h_{\min} and represents the set of pixels belonging to the background. The Max-Tree of a 1D signal is shown in Fig. 6.2.

Each Max-Tree node except for the root, points to its parent at level $h' < h$. The root node points to itself. This linking property simplifies the computation of peak component attributes since every parent inherits the auxiliary data stored in children nodes along the same root-path. In the case of increasing attributes such as area or volume, inheritance is a simple accumulation while for the more complicated case of shape attributes like the one in (6.8) the process relies on more sophisticated attribute handling functions described in [57, 85].

The construction of the tree is done recursively. A flooding function fed by a set of hierarchical first-in first-out (FIFO) queues, upon receiving a pixel updates the auxiliary data buffer of the node currently being flooded. It then inspects its neighbors and places them in the appropriate queue entries. If a neighboring pixel is at a higher level $h' > h$, flooding the current node at h pauses and a new function call initiates the process at h' . Once a node is fully flooded, i.e. there are no more pixels in the queue for that level, it is finalized by detecting its parent, updating its members and setting the appropriate flags. The function returns the auxiliary data to the parent node and flooding continues or initiates at that level. The process terminates when flooding the root node is completed. Implementation details are discussed in [55, 65].

The filter function, realized in a separate stage, reads the tree structure by visiting each node separately starting from the root. For every node an auxiliary data interpreter computes the attribute value which in turn is compared against the pre-specified attribute threshold λ . Nodes failing the criterion are removed by lowering the gray-level of their member pixels in accordance to the filtering rule chosen. More on these strategies are given in [19]. Restitution simply assigns the new levels to the corresponding pixels.

6.4.2 The k -subtractive Implementation

The k -subtractive filtering rule defined in Sec.6.3.2 is implemented as a separate function that takes as input a Max-Tree structure, an auxiliary data interpreter and the two parameters λ and k . It requires a single pass of the tree in which each node is visited just once.

In this version of the Max-Tree two extra fields per node are required: *PeakLevel* and *kprime*, which store the gray level of the descendent with the highest gray level, and the propagation range respectively. The first is initialized to each node's original level, and the second to 0 while constructing the tree. Prior to filtering the *SetPeakLevels()* function is

Algorithm 4 The k -Subtractive filtering rule.

```

process  $k$  - subtractive(MaxTree  $t$ , void * Attribute,  $\lambda$ ,  $k$ )
  build Max-Tree for an image  $f$ 

  var  $idx$  ,  $parent$  ;
  var  $node$  ,  $parnode$  ;
  var  $difflev$  ;

  /* process root */
  if /  $root \rightarrow Attribute > \lambda$  then
     $root \rightarrow kprime = k$  /* preserve and set maximal  $k$ -restoration level */
     $root \rightarrow NewLevel = root \rightarrow Level$ 
  else
     $root \rightarrow kprime = 0$ 
     $root \rightarrow NewLevel = 0$ 
  endif
  for all levels  $l$  starting at  $h_{min} + 1$  to  $h_{max}$  do
    for all nodes at each level  $l$  do
      compute node index  $idx$ 
      find node's parent  $parent$ 
       $difflevel = node \rightarrow Level - parnode \rightarrow Level$ 
      if  $node \rightarrow PeakLevel - parnode \rightarrow Level > k$  AND  $node \rightarrow Attribute > \lambda$  then
        /* preserve and set maximal  $k$ -restoration level */
         $node \rightarrow NewLevel = parnode \rightarrow NewLevel + difflevel$ 
         $node \rightarrow kprime = k$ 
      else
        if  $difflevel > parnode \rightarrow kprime$  then
           $node \rightarrow NewLevel = parnode \rightarrow NewLevel + parnode \rightarrow kprime$ 
           $node \rightarrow kprime = 0$ 
          /*  $k$ -restoration completed */
        else
           $node \rightarrow NewLevel = parnode \rightarrow NewLevel + difflevel$ 
           $node \rightarrow kprime = parnode \rightarrow kprime - difflevel$ 
          /* remainder to be  $k$ -restored */
        endif
      endif
    endfor
  endfor
  for all pixels  $p$  restitute the filtered image

```

called. This routine traverses the tree from the leaves to the root, and at each node sets the *PeakLevel* field of the parent to its own *PeakLevel*, if its *PeakLevel* is higher than the parent's *PeakLevel*. This ensures that the *PeakLevel* field of each node C_h^i is set to the maximum within the corresponding peak component P_h^i .

The *PeakLevel* is used to determine whether each node examined associates to a k -peak component or not. That is, if the difference between the node's *PeakLevel* and its parent's level is greater than or equal to k then there is sufficient contrast range and the corresponding P_h defines a $P_{h,k}$. The *kprime* member specifies the propagation range, i.e. which nested peak components are to be preserved independent of their attribute measure if there exists a k -peak component among their ancestors.

The function starts by reading all nodes from the root upwards. The root node is handled separately since does not carry any restore decisions from previous nodes. If it defines a k -peak component that satisfies Λ it sets *kprime* to the maximum range i.e. k , otherwise to 0, as is its *NewLevel* field. After processing the root, the rest of the nodes are scanned from root upwards. If the meet criterion Λ , their *NewLevel* and *kprime* fields are set as in the case of the root. If criterion Λ is not met, or if the difference between their original level h and *PeakLevel* is smaller than k there are two situations. If the difference between its gray level and its parent's is greater than the propagation range *kprime* of its parent (which may vary from 0 to k) then it must be lowered to a new level which is that of its parent plus the parent's propagation range. Since it is a rejected component which is out of range, its *kprime* is set to 0, i.e. it does not carry any restore decisions from its ancestors because its out of their range and also has nothing to propagate itself. If however it is within the propagation range of some ancestor, it is preserved and updates its level to that of its parent plus the gray levels difference with it, and propagates the remaining range further up. The pseudo-code for the k -subtractive filtering function is given in Alg. 4.

The process terminates when all nodes are visited. The output image is then restituted based on the new levels of the nodes. The image restitution is the same as with the earlier implementation of the subtractive rule. Variants for the direct, min and max rules can be made in a similar way.

6.5 Applications on Volumetric Data and Discussion

In this section we demonstrate the performance of 3D shape filters on k -level hyperconnected volume sets and compare our findings against the outputs of the same filters configured with standard connectivity. In the experiments that follow we employ the non-compactness filter of Section 6.2.2 configured with the k -subtractive filtering rule of the previous section. One of the data sets is handled with the recently introduced *Sparseness* attribute of [90] for a better illustration of the method we are presenting. The results are displayed using iso-surface projections or direct volume rendering (DVR) based on color tables of the alpha component.

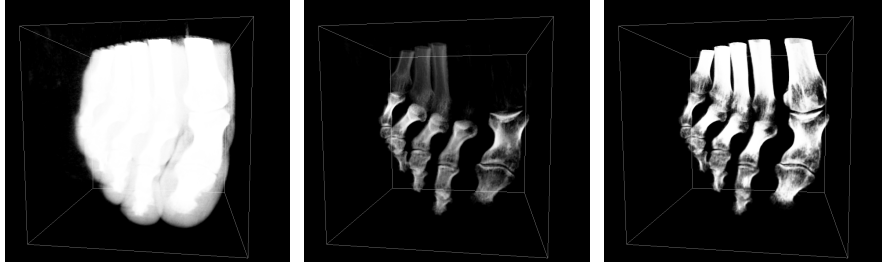


Figure 6.3: Foot - (left to right) The original volume set in color table projection using alpha components; the output of the non-compactness filter using standard connectivity; the output of the same filter with k set to 120.

The latter is global factor to change the overall transparency of the object independent of the data value. Computation times are reported for each set separately and algorithm dependencies are discussed in the last subsection. All experiments were carried out on an Intel Pentium 4, 3.2 GHz CPU with 2GB memory.

6.5.1 The foot data set

The first data set shown in Fig. 6.3 is a rotational C-arm X-ray scan of a human foot, courtesy of Philips Research, Hamburg, Germany. The data are in the form of a $256 \times 256 \times 256$, 8-bit volume and the objective is to enhance the bone by suppressing the tissue. We use the non-compactness filter for this purpose with λ set to 1.2 and visualize the result using color-table projection configured with alpha components. Selecting a higher λ removes much of the bone structure irrecoverably while for lower values much of the tissue remains in the form of elongated noise patterns. The filter output using standard connectivity fails to retain the integrity of the bone and parts of it like the upper half of the first two toes is removed leaving only a few elongated components that satisfy the criterion. Moreover in iso-surface projection (not shown here) one can see low contrast elongated tissue fragments resting on the background. These fragments being of low contrast can be removed with a relatively low value for k , but a large enough value like $k = 120$ also allows the recovery of the missing bone leading to the result shown in the bottom image of Fig. 6.3.

6.5.2 The CT-Knee data set

The second data set shown in the first column of Fig. 6.4 is a CT scan of a knee with an anterior tibial osteotomy, courtesy of the Department of Radiology, University of Iowa. The data are in the form of a $379 \times 229 \times 305$, 8 bit volume and the objective like in the previous case is to enhance the bone by suppressing both the tissue and the supporting badges shown

in the top image at iso-level 1. We use the non-compactness filter with λ set to 0.5 for the same reasons as before. Though most of the tissue is removed (at iso-level 12 it is not visible at all), the filter output using standard connectivity cannot remove parts of the badge which are elongated enough to satisfy Λ . Setting the iso-level higher clears the volume but also removes parts of the bone joints which should be visible. Since the bone remains intact, setting $k = 40$ is sufficient to remove the remaining badge fragments which are of lower contrast. The final result is shown at the bottom image of the first column.

6.5.3 The MRI-Head data set

The third data set shown in the second column of Fig. 6.4 is an MRI scan of a human head, courtesy of the Computer Graphics Group, University of Erlangen, Germany. The data are in the form of a $256 \times 256 \times 256$, 8 bit volume and the objective is to enhance the exterior of the head leaving the skin details intact. We use the sparseness attribute of [90] for this purpose with a small value for λ . The top image of the second column shows an ortho-slice of the volume set where the noise surrounding the head is visible. The sparseness filter using standard connectivity with $\lambda = 2$ fails to remove the noise efficiently and we see that at lower levels where the particles have a high sparseness measure, the problem remains. Using a high enough value for k , set to 100 in this case, eliminates all noise while restores bright point-size components on the skin surface. The result is shown at the bottom image of the second column and can also be achieved using the flatness attribute of [90]. We chose the specific filter though because it demonstrates clearly the features of the proposed method.

6.5.4 The CT-Chest data set

The fourth data set shown in the first column of Fig. 6.5 is a CT scan of a female chest, courtesy of the Department of Radiology, University of Iowa. The data are in the form of a $384 \times 384 \times 240$, 8 bit volume and the objective is to enhance the skeleton by suppressing the tissue and the surface on which the subject rests on during the tomography. The top image shows the original volume at iso-level 40. The non-compactness filter output using standard connectivity clears the volume sufficiently (middle image) even at very low intensities (iso-level 3 in this case) but fails to remove the resting structure parts of which are visible on the side body and on the back of the skeleton. Since the resting structure appears de-touched from the filtered set a high enough value for k , higher than the contrast of the targeted structure, removes it together with all previously surviving details other than the skeleton. The result is shown at the bottom image of the first column.

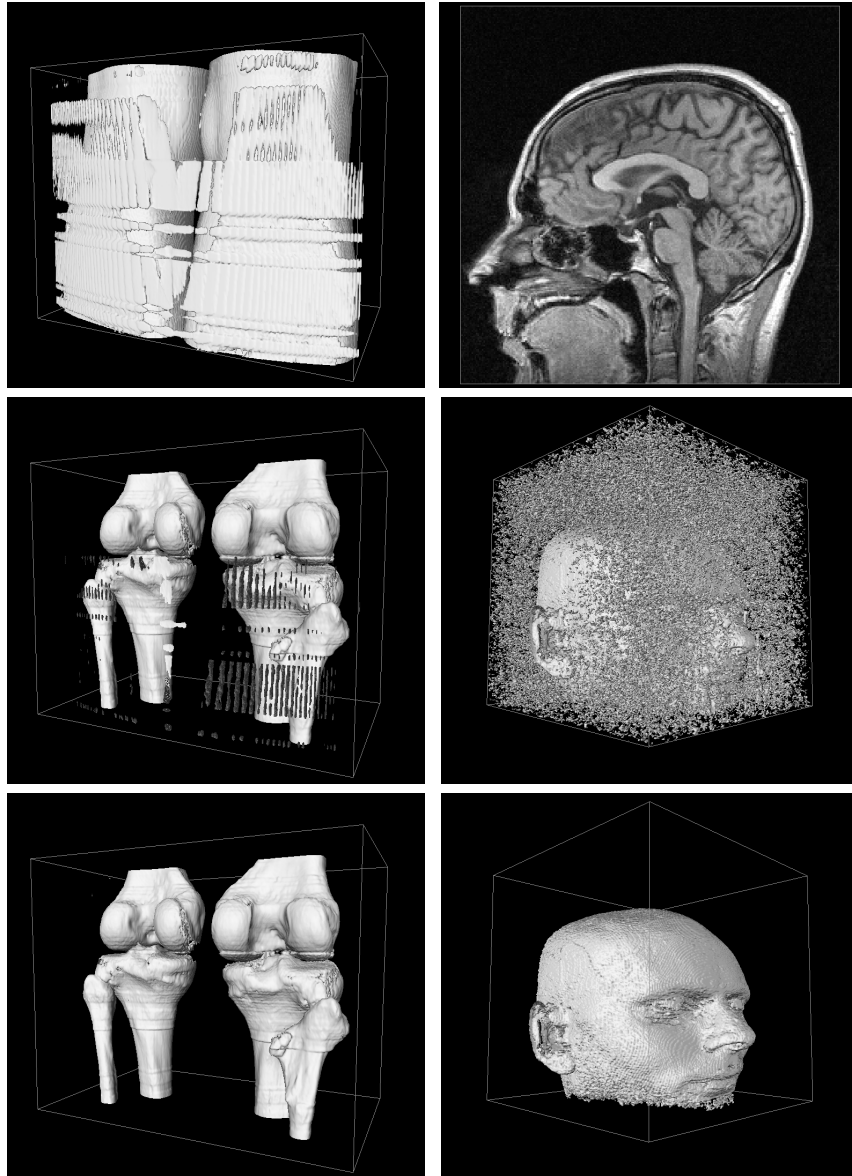


Figure 6.4: CT scan of a knee (first column, top to bottom); The original volume set at iso-level 1; the output of the non-compactness filter using standard connectivity and the output of the same filter setting k to 120, both at iso-level 12. MRI scan of a head (second column, top to bottom); The original volume set in color-table projection; the filter output using standard connectivity and setting k to 100, both at iso-level 1.

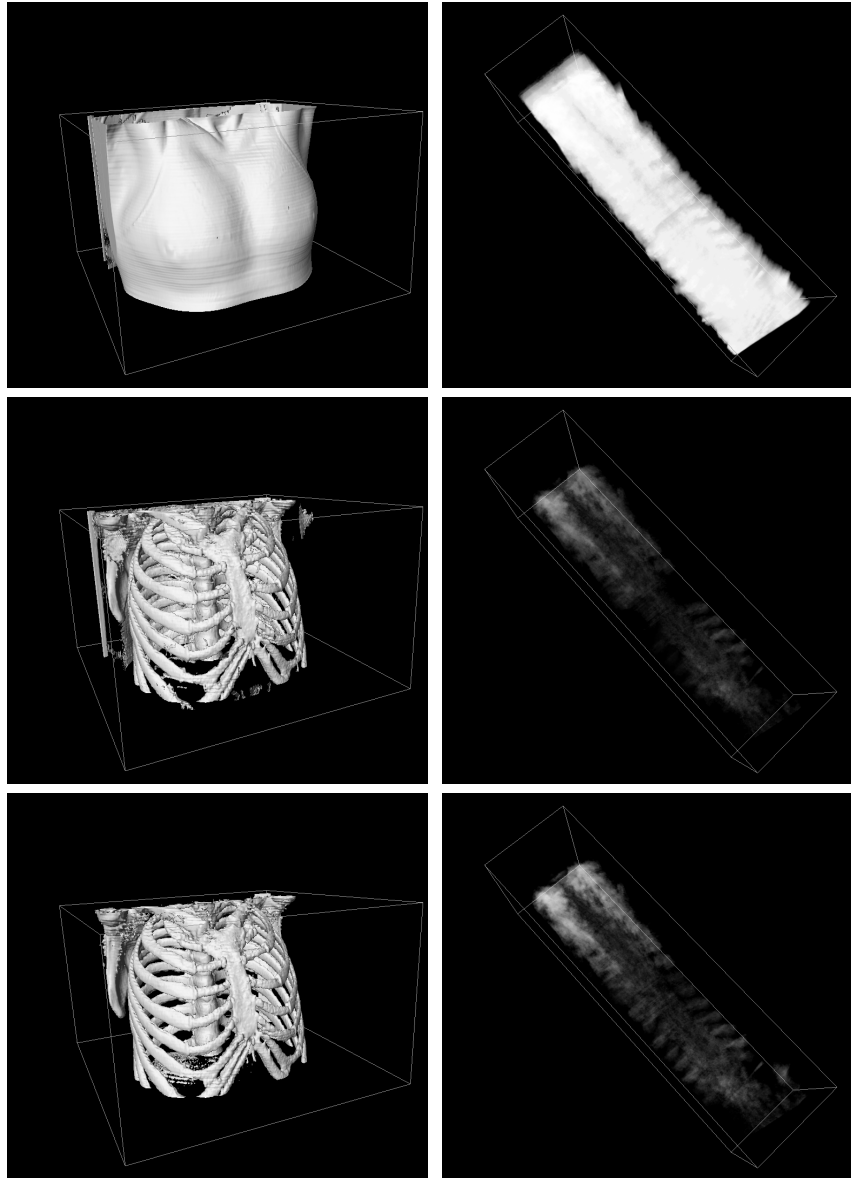


Figure 6.5: CT scan of a woman's chest (first column, top to bottom); The original volume set at iso-level 40; the output of the non-compactness filter using standard connectivity and the output of the same filter setting k to 110, both at iso-level 3. A 3D ultrasound of the human spine (second column top, to bottom); The original volume set in color-table projection; the non-compactness filter output using standard connectivity and setting k to 5.

6.5.5 The spine data set

The last data set shown in the second column of Fig. 6.5 is a 3D ultrasound of a human spinal cord, courtesy of K.E. Purnama, Institute of Mathematics and Computing Science, University of Groningen, The Netherlands. The data are in the form of a $130 \times 161 \times 490$, 8-bit volume and the objective is to de-noise the set making the ribs and parts of the vertebra visible to the degree possible. Certain ribs appear as point clouds due to the acquisition method, making them vulnerable to most attribute criteria. The non-compactness filter using standard connectivity and $\lambda = 2$ retrieves much of the structure but fails to capture these ribs which are treated as noise and get removed. Setting $k = 5$, the filter allows for much of these point clouds to get clustered to meaningful structures while letting the noise form detached low elongation and low contrast structures which are removed. The result is shown at the bottom image of the second column. The same happens with the core bone tissue in the upper part of the image preventing removal of peak components at higher levels and thus further reduction in contrast.

6.5.6 Parameter Selection and Computational Complexity

The computational complexity of the Max-Tree algorithm is discussed in [65]. It has a strong dependency on the image content which affects the size of the tree thus the number of recursions while flooding, and the number of nodes to be processed while filtering. Once the tree structure is completed, the *SetPeaks()* function is called if $k > 0$. This is an extra pass through the tree structure which again is content dependent and contributes a fixed time overhead. Similarly, the k -subtractive, just like the ordinary subtractive rule, requires only one pass of the tree structure involving a few additional **if – else** statements. It is content dependent, through its dependence on the number of nodes, as well and totally independent of both λ and k . In Table 6.1, we list the CPU timings for the data sets presented in the previous subsection. To account for differences between older versions of the Max-Tree code and the present, timings for the standard connectivity are given using this latest version and setting $k = 0$. Worst case complexity for the flooding phase is $O(GN)$ with G the number of gray levels, and N the number of pixels. For *SetPeaks()* and the filtering stage we have a worst-case complexity of $O(N)$, because no more Max-Tree nodes than pixels can exist.

The parameter k is very much depended on the data-set. To remove low contrast but of high attribute measure structures which rest on the background, usually small values are sufficient. If these structures are within larger ones that should also be removed, together with an appropriate value for λ a high enough value for k is required that exceeds the maximum span of gray levels of the targeted objects. This is to ensure that for any component which satisfies the filter criterion there is not enough contrast to define a k -peak component. To recover components that would be removed by a filter relying on standard connectivity, a

Table 6.1: CPU timings for each data set (in sec.).

Data Set	Attribute filter relying on:				
	Standard Conn.		Hyperconn.		
	<i>build</i>	<i>filter</i>	<i>build</i>	<i>filter</i>	<i>k</i>
CT Foot	14.206	0.209	13.828	0.113	120
CT Knee	24.582	0.175	24.588	0.142	40
MRI Head	19.039	0.642	19.022	0.233	100
CT Chest	30.082	0.208	30.195	0.187	40
US Spine	7.731	0.083	7.490	0.065	5

high enough value for k is required such that there exist k -peak components that satisfy the criterion less than k levels below them. For $k = 0$, the k -peak components become the regular peak components of the image and the notion of k -level hyperconnectivity essentially reduces to the standard connectivity.

6.6 Conclusions

In this paper we presented a method for computing attribute filters that rely on a notion of hyperconnectivity instead of the standard connectivity. It is shown that the properties of connected attribute filters carry over to the hyperconnected case. We then focused on k -flatzones to provide our hyperconnectivity. The aim was to involve contrast-based information on the filter's decision making. Filters based on this strategy can reject high attribute measure structures that are of low contrast by controlling an additional parameter k . Similarly, they can preserve fine details that fail the filter's criterion if found in a high contrast region. The benefits of this can be seen on the results of our experiments for which we used data sets that regular filters fail to enhance properly.

Hyperconnectivity [9, 72] is a recent introduction in the theory of connectivity and there are open challenges in both the theoretical and practical sides of it. Most important are the axiomatics of the families of operators which return hyperconnected components marked by the points $x \in E$. Due to the dependency of our method on a "base" connectivity class, we bypassed this limitation by making use of regular connected components. This approach however cannot be generalized and the problem remains to be solved. Note that there are no restrictions on the nature of this base connectivity class meaning that second generation and other derivatives of standard connectivity may also be used. Moreover, we see that \mathcal{H} becomes a hyperconnectivity class for $k > 0$. If $k = 0$ the expression for \mathcal{H} yields a standard connection as it should be.

The hyperconnected attribute filters were implemented on the Max-Tree structure in the

form of a filtering rule. This as explained earlier, allows setting k interactively and has a great impact on the performance of the algorithm when compared with the alternative method using the dual-input Max-tree. A demo program together with some sample data sets and the source code are available in <http://www.cs.rug.nl/~michael/MTdemo/>. The CPU timings reported for each data set show that the overhead of the new method deviates less than 1% from that of regular connected attribute filters and the algorithm is linear with size.

In future work, we are looking into the theory of hyperconnectivity aiming to formalize an operator capable of returning hyperconnected components. This is an essential step for exploring the field of hyperconnected morphology and its operators. Moreover, we are looking at the benefits of customizing the hyperconnectivity class we presented by selecting different types of base connections.

7.1 Conclusions

THIS thesis describes three extensions to the theory of connectivity in mathematical morphology. Connectivity plays a key role in the development of robust and efficient tools and operators for both image analysis and processing. The work presented and analyzed in this book is not limited to the theoretical aspects of connectivity only, but emphasizes on the practical issues of it as well. In each one of the three extensions, together with the mathematical background, sets of operators are given based on which attribute filters can be designed. Aiming at transferring these methodologies to real world problems, each thematic section is accompanied with an appropriate algorithm that implements the developments presented. Experiments that demonstrate the features, capabilities and limitations of the proposed filters and operators are also given in each section. CPU timings and efficiency evaluations together with comparisons to other methods in some cases, aim at giving the reader the opportunity to draw personal conclusions on the usability of each framework and suitability for integration to other external modules. The overall work covered in this thesis breaks down to five chapters which are briefly summarized below.

Chapter 2 presents the concept of mask-based second-generation connectivity in which the previously individual clustering-based and contraction-based connectivities are now hosted. Connectivity openings associated to this new type of second-generation connectivity make use of mask images rather than structural operators, eliminating dependencies on their properties. The connectivity of images in question is no longer dictated by the choice of operator or the size of the structuring element used but instead on the mask patterns. Masks can be generated in custom ways to meet the demands of any given problem. Application examples with operators previously not supported are filters acting on image pairs of the same scene taken in different frequency bands (e.g., optical/IR combinations) or using different imaging modalities (optical/range imaging or registered CT/MRI pairs).

Attribute filters configured with mask-based or regular second-generation connectivity can be computed efficiently using the dual-input Max-Tree (DIMT) algorithm introduced in Chapter 2. This tree-based image representation algorithm operates on both 2D and 3D images and supports a wide range of different attributes. Experiments on this new framework

demonstrate its clear advantages against regular clustering based operators. To gain a further insight into the parameters influencing the performance of clustering-based operators, Chapter 3 deals with this issue exclusively. Apart from the obvious effect of the attribute threshold, the role of the structural operator chosen and the size of the structuring element used, in the case of regular second-generation connectivity, are investigated together with the way attributes are computed. The work is complemented with a concurrent implementation of the dual-input Max-Tree algorithm, presented in Chapter 4. The method involves an intuitive parallelization strategy recently introduced for regular Max-Trees, that is based on the Union-Find algorithm for efficient merging of the subtrees - one for each thread. The speedups reported and experiments on 3D data suggest that the DIMT algorithm can handle large data sets efficiently and at reasonable timing.

The problem of oversegmentation, a condition appearing in both regular contraction-based and mask-based connectivity, is dealt with in Chapter 5. Starting from Serra's original work on image partitions, a new type of connectivity class, the π -connection, was presented. The definition involves classes of image partitions which are used in ways analogous to masks by mask-based second-generation connectivity. The π -connectivity opening presented, differs from its mask-based equivalent in that it handles pathwise connected image regions previously treated as sets of singletons, as independent entities to which meaningful attributes can then be assigned. Though this resolves in part the oversegmentation problem in practical applications, the π -connected operator is shown to be limited. That is because extensions to gray-scale violate certain hierarchical ordering properties required by the existing filtering rules. Bypassing this remains a topic for further investigation. It is possible though to compute gray-scale connected pattern spectra by restating certain assumptions. A brute force algorithm is given and a set of experiments on texture-based image classification of diatoms proves that the method is more reliable compared to regular contraction-based connected pattern spectra and in some cases outperforms regular connected pattern spectra based methods too.

The last chapter, Chapter 6 touches upon the notion of hyperconnectivity. Developments in this field were minor primarily due to the fact that there is no operator up to date that given a point on the image returns a hyperconnected set of maximal extent. It is shown though, that in gray-scale image analysis and under certain conditions, this can be bypassed making use of k -flat zones. This feature was used to define attribute filters involving contrast information together with structural characteristics. The associated operators are configured with a "base" connectivity class from which a combination of nested connected components and flat zones can be extracted as individual hyperconnected sets of maximal extent.

The work was motivated by the loss of fine, bright structures usually observed at higher levels that usually fail the regular filter's criterion. Employing this scheme, it is now possible to retain such structures assuming they rest on a high contrast region of the image that meets the criterion. That, in a sense, suggests that they are part of what rests below them. By con-

trast, undesired structures resting on the background that would previously be accepted by the filter are now suppressed. In both cases the contrast range is controlled by the parameter k which can be set interactively during the filtering stage of the algorithm, that comes along with this theory. The algorithm was designed intentionally like that to avoid rebuilding the tree-based image representation structure for each new value of k . Experiments show that the method which can complement any existing attribute filter, and yields far better results when compared with those of the same filter configured with standard connectivity. Moreover, setting $k = 0$ reduces the hyperconnected to a regular connected filter and this allows for easy comparisons between the two methods.

7.2 Future Work

The introduction of mask images in second-generation connectivity eliminated constraints imposed from earlier frameworks on the ways an image can be connected. Masks can be generated from any arbitrary operator and need not to originate from the input image. From the experimentation on protein images in Chapter 2 it became apparent that involving directionality/orientation as a criterion for mask generation can resolve a number of problems such as noise clustering and leakage. Though in 2D this is feasible, attempts to carry out similar experiments in 3D showed that the time overhead was prohibiting. Deriving efficient 3D mask generation algorithms that make use of such criteria remains an open challenge. Efficient steerable filters which need not be morphological, and spatially variant mathematical morphology [4, 5] might also be of use here. Work is also taking place on similarity criteria for clustering purposes.

Cases involving contractions are also under investigation. Though in Chapter 5 it was shown that oversegmentation can be countered in part with the aid of π -connected operators there remains the need of an efficient algorithm that computes attribute filters and pattern spectra configured with this type of connectivity in gray-scale. Concerning the diatom classification problem, it is believed that different classifiers using the existing feature vectors may increase the success rates further.

Oversegmentation may also be countered through reconstruction. Masks generated by anti-extensive structural operators usually contain compact structures (stable components) which can be used as seeds to retrieve the desired missing details. The challenge here is finding appropriate shape attributes to limit the reconstruction operators to the regions of interest. This is related to the work on reconstruction criteria [80] and viscous lattices [71].

The field of hyperconnectivity, being relatively new, offers many challenges both in theory and in algorithmics. It is seen as priority to formalize an operator capable of returning hyperconnected components. This is an essential step for exploring hyperconnected morphology and its operators. The specific type of hyperconnectivity presented in Chapter 6 though limited, allowed the introduction of the first hyperconnected attribute filter, which

was shown to deliver promising results. Experimentation on this field is taking place to exploit the benefits of customizing the hyperconnectivity class presented with different types of base connections.

Both in connectivity and hyperconnectivity, we have only studied anti-extensive filters. Though the extensive counterparts follow by duality, auto-dual filters [13, 48, 50, 60], and beyond [75] are not trivially handled by the techniques developed in this thesis. However in the auto-dual case it is plausible that a combination of a dual-input Max-Tree and Min-Tree could yield a second-generation connected equivalent to the level-line tree of [49].

Connected and hyperconnected morphology have strengths that make them competitive to many other image analysis and processing methods. It is an active field of research that finds use in many modern computer vision applications including medical imaging. This thesis has contributed new ways of grouping pixels together in meaningful ways to represent objects more robustly.

Samenvatting

DIT proefschrift beschrijft drie uitbreidingen van de theorie van connectiviteit in de mathematische morfologie. Connectiviteit speelt een belangrijke rol bij de ontwikkeling van robuuste en efficiënte methoden en operatoren voor beeldbewerking en beeldanalyse. De resultaten in dit proefschrift zijn niet beperkt tot nieuwe theoretische aspecten van connectiviteit, maar benadrukt praktische aspecten evenzeer. In ieder van de drie uitbreidingen presenteren naast wiskundige theorie ook we diverse operatoren op basis waarvan attribuutfilters kunnen worden ontworpen. Om problemen in de praktijk te kunnen aanpakken, worden in alle gevallen efficiënte algoritmen afgeleid en gepresenteerd. Verschillende experimenten, die de sterke en zwakke punten van de nieuwe methoden demonstreren worden ook beschreven. De efficiëntie van alle algoritmes is geëvalueerd, zowel door complexiteitsanalyse als door middel van CPU timings. Hiermee krijgt de lezer eenvoudig inzicht in de toepasbaarheid van de nieuwe methoden voor zijn eigen specifieke beeldbewerkingsprobleem. Dit proefschrift omvat vijf belangrijke delen die hieronder zijn samengevat.

Hoofdstuk 2 beschrijft het concept van masker-gebaseerde tweede-generatie connectiviteit, waarin de voorheen afzonderlijke clustering-gebaseerde en contractie-gebaseerde connectiviteiten kunnen worden verenigd. Connectiviteits-openingen geassocieerd met deze nieuwe vorm van connectiviteit gebruiken een masker beeld in plaats van morfologische operatoren gebaseerd op strurerende elementen, en zijn daardoor niet afhankelijk van de eigenschappen hiervan. Het is dus niet meer de keuze voor een bepaalde operator die de connectiviteit vastlegt, maar de patronen in het masker beeld. Maskers kunnen met op een veelzijdiger manier worden gekozen om aan de vereisten van verschillende beeldbewerkingsproblemen te voldoen. Ook is het mogelijk om masker beelden te gebruiken die simpelweg opnamen zijn van hetzelfde object of dezelfde scene met bij een andere golflengte (b.v., zichtbaar licht/IR combinaties), of met een ander apparaat (b.v., zichtbaar licht/LIDAR, of CT/MRI combinaties).

Attribuutfilters geconfigureerd met zowel de masker-gebaseerde als de operator-geba-

seerde tweede-generatie connectiviteit kunnen efficiënt uitgerekend worden met behulp van het dual-input Max-Tree (DIMIT) algoritme dat in Chapter 2 wordt geïntroduceerd. Deze beeldrepresentatie werkt zowel in 2D als in 3D en kan filteren gebaseerd op een groot aantal attributen (object eigenschappen).

Experimenten met de nieuwe vormen van connectiviteit laten een duidelijk voordeel ten opzichte van clusterende connectiviteit zien. In Hoofdstuk 3 wordt de invloed van parameterinstellingen op de effectiviteit van deze filters onderzocht. Afgezien van de attribuut-instellingen, worden de rollen van de structurele operator en de grote van het structurerende element onderzocht, in het geval van operator-gebaseerde second-generatie connectiviteit. Ook worden twee verschillende manieren van et berekenen van de attributen getest. Tot slot van dit onderdeel wordt in Hoofdstuk 4 een parallelle versie van het DIMIT algoritme gepresenteerd. Deze variant werkt door Max-Tree structuren voor disjuncte delen van het beeld (of volume), en deze dan efficiënt te recombineren in één enkele boom-structuur. De gemeten versnelling laat zien dat hiermee grote datasets snel verwerkt kunnen worden.

Het probleem van oversegmentatie, dat optreedt in zowel reguliere contractie-gebaseerde en masker-gebaseerde connectiviteit, wordt behandeld in Hoofdstuk 5. Uitgaande van Serra's originele werk op het gebied van beeld partities, wordt een nieuw type connectiviteits klasse, de π -connectie, gepresenteerd. De definitie is gebaseerd op de partitie-classes in een beeld die gebruikt worden op een vergelijkbare manier als de maskers in masker-gebaseerde connectiviteit. De π -connectiviteits-opening verschilt zodanig van de masker-gebaseerde tegenhanger dat pad-samenhangende beeld-onderdelen die voorheen opgebroken werden in individuele pixels (of voxels) nu wel samenhangend blijven, en dus van zinvolle attributen kunnen worden voorzien. Hoewel verschillende problemen op deze manier worden opgelost, laten we ook zien dat operatoren gebaseerd op π -connectiviteit gelimiteerd zijn. Dit komt vooral omdat efficiënte uitbreiding naar grijswaarde beelden niet mogelijk zijn, omdat niet aan de vereiste hiërarchische ordenings-eigenschappen voor de huidige attribuut filters voldaan kan worden. Het omzeilen van dit probleem vergt nog meer onderzoek. Ondanks deze problemen is het wel mogelijk om een pseudo patroon-spectrum in grijswaarde beelden uit te rekenen. Een brute-kracht algoritme wordt gepresenteerd en experimenten laten zien dat textuur-gebaseerde identificatie van diatomeeën beter gaat met de nieuwe methoden gebaseerd op π -connectiviteit, dan met contractieve, masker-gebaseerde connectiviteit. In sommige gevallen is de nieuwe aanpak ook beter dan traditionele connectiviteit.

Het laatste hoofdstuk, Hoofdstuk 6 beschrijft een nieuwe vorm van hyperconnectiviteit. Tot nu toe waren er niet veel ontwikkelingen in dit gebied met name omdat er geen operator is ontwikkeld die gegeven een punt in beeld, de hyperconnectiviteits componenten vindt. Het wordt aangetoond dat in bepaalde gevallen in grijswaarde beelden onder omstandigheden dit probleem omzeild kan worden, door gebruikt te maken van z.g. k -flat zones. Hiermee kunnen we attribuut-filters ontwikkelen die zowel attribuut informatie als contrast informatie benutten. De geassocieerde operatoren worden geconfigureerd met een "basis"connectiviteit

waaruit combinaties van geneste componenten worden gemaakt en gebuikt als individual hyperconnectiviteits componenten.

Het werk is geïnspireerd door het verlies van fijne, heldere structuren die vaak worden gevonden binnen structuren die wel door het filter heen komen, maar zelf door het filter worden verwijderd. Met de nieuwe methode is het nu mogelijk deze structuren te behouden, als ze onderdeel zijn van een voldoende contrastrijke object dat wel voldoet aan de filter criteria. Ze worden dus als onderdeel van een groter geheel beschouwd. Tegelijkertijd worden zwakke (veelal ruis) structuren die vroeger vanwege hun vorm door het filter heen kwamen onderdrukt als ze op een verder uniforme achtergrond staan. In beide gevallen bepaald een parameter k de contrast-omvang die gebruikt wordt. Deze parameter kan interactief ingesteld worden in het nieuwe algoritme dat hiervoor werd ontwikkeld. Dit kan omdat dezelfde Max-Tree structuur voor iedere willekeurige waarde van k kan worden (her)gebruikt. Experimenten laten zien dat deze nieuwe hyperconnectiviteit ingebed kan worden in vele bestaande attribuut-filters, en dat op veel 3D data een veel beter resultaat wordt bereikt dan met gewone connectiviteit. Bovendien reduceerd bij $k = 0$ het algoritme tot die voor de gewone connectiviteit wat vergelijking van de methoden erg eenvoudig maakt.

Publications

Parts of the work presented in this thesis have been published in scientific journals and conference proceedings. Chapters 5 and 6 have been submitted to *Pattern Recognition* and *IEEE Trans. Pattern Analysis and Machine Intelligence*, respectively.

Scientific Journal Paper

- G.K. Ouzounis and M.H.F. Wilkinson. Mask-based second generation connectivity and attribute filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, June 2007, pp. 990-1004

Full papers in conference proceedings

- G.K. Ouzounis and M.H.F. Wilkinson. A parallel implementation of the Dual-Input Max-Tree algorithm for attribute filters. *Int. Symp. Math. Morph. (ISMM) 2007*, Oct. 10-13, 2007, Rio de Janeiro, pp. 449-460
- G.K. Ouzounis and M.H.F. Wilkinson. Filament enhancement by non-linear volumetric filtering using clustering-based connectivity. *Lecture Notes in Computer Science*, Vol. 4153, *International Workshop on Intelligent Computing in Pattern Analysis/Synthesis, IWICPAS 2006*, (N. Zheng, X. Jiang, and X. Lan, Eds.), Xi'an, China, August 2006, pp. 317-327.
- G.K. Ouzounis and M.H.F. Wilkinson. Countering over-segmentation in partitioning-based connectivities. *Int. Conf. Image Proc. 2005*, Vol. 3, September 11-14, Genova, Italy, pp. 844-847.
- G.K. Ouzounis and M.H.F. Wilkinson. Second-order connected attribute filters using Max-Trees. *Mathematical Morphology: 40 years on, Proc. Int. Symp. Math. Morph. (ISMM) 2005*, pp. 65-74.

Submitted Papers

- G.K. Ouzounis and M.H.F. Wilkinson. Hyperconnected attribute filters based on k-flat zones for 3D medical imaging. Submitted to IEEE Trans. Pattern Analysis and Machine Intelligence, 2008.
- G.K. Ouzounis and M.H.F. Wilkinson. Partition-induced connections and operators for pattern analysis. Submitted to Pattern Recognition, 2008.

Acknowledgments

Scientific research requires an open mind, clear of all restrictions that one learns and accumulates from everyday life. Tommy Dewar once said that “minds are like parachutes, they only function when they are open.” Judging my progress I feel that it is exactly there where dr. Michael Wilkinson, my supervisor, mentor and friend, has contributed the most. Michael has offered me the opportunity to prove myself through an intense work schedule which led to many exciting discoveries. His broad knowledge on a variety of topics inspired me in looking beyond the strict limits of our science. Loud discussions and arguments “ala mediterranea”, which were often misunderstood as war cries by several members of staff, were the source of all the scientific recipes that emerged. In life outside the university, both he and his wife Meiny stood by my side from the very first day of my stay in Holland (not to mention the fantastic dishes they prepared!!!). I thank you both a lot for all the things you did for me.

In the same spirit, I would like to thank my promotor Prof. Nicolai Petkov who assisted me in several ways including financial support for conferences and other events. Nicolai just like myself, appreciates good humor and that relaxes the working conditions. He is also a fan of delicate cuisine and I bet that is why he let me organize the group dinners. I would also like to acknowledge the contribution of dr. Michael Biehl in matters concerning intelligent algorithms and statistical learning. Though we did not have the opportunity to collaborate in a common project, Michael suggested a whole lot of interesting ideas and answered many of my questions in this field through a number of productive conversations and meetings.

Together with the members of the academic staff, I would also like to thank my colleagues, Aree Witoelar and his wife Cynthia, Lidia Sanchez, Florence Tushabe, Easwara Subramanian, Giuseppe Papari, Petra Schneider, Alle Meije Wink, Anarta Ghosh, Koen de Raedt, Michael Ten Caat, Ronald van den Berg and Alessandro Crippa. I specially thank Erik Urbach, Andrei Jalba and Michel Westenberg for their valuable contributions in source code, bug fixing and knowledge sharing. Without them I would be still lost in the Max-Tree jungle!

I am also expressing my gratitude to the secretarial staff of our department and more

specifically to Desiree Hansen, Ineke Schelhaas, Helga Steenhuis, Annette Korringa, Yvonne van der Weerd and Janieta de Jong. These ladies backed me up in countering the paperwork in a language I never managed to learn properly. A very very special thanks goes to Esmee D. Elshof, both a friend and a colleague who stood by my side in difficult times. Esmee, I honor your friendship and thank you for everything.

In the same way these ladies helped me with something I was totally lost with, Harm Paas and Jurjen Bokma helped me with Linux!!! Gentlemen thank you for your support and I apologize for trashing my linux box so often.

A great thank you goes to all my friends in Groningen with whom I have four wonderful years to remember. Mentioning them all would probably occupy more space than the scientific material itself, thus I would like to constrain the list to the following: my good friend Dimitris Arvanitis, Vassilis Linardos, Panagiotis, Giannis Vogeas, Grigoris and Nikoletta Vogeas, Evelina to kourkoutaki, Elenitsa Tsopanidou, the staff of La Femme Grecque and in general the Greek "clan" of Groningen and my friends in the Greek society of north Holland; my good friend Nico Gonzalez and his wife Marianna, Toma, Pasqual, Fred and the "petrol gang" of Groningen; the boys of the Equilibrium Delirium group, that is Mo, Rob, Azaria, and Aree; my special "balkano" friend Saša Martić and his sweet Olga, Paris Avgeriou and Sofia, Andrei and Cosmina Jalba, Alex Crippa and Ruut, Sophie van de Gevel, Smaranda Andreas and Adriana Krawczyk. A special thanks goes to my sweet Saskia Berrelkamp and her parents. Thank you for all your help, your warm hospitality and for all the other things you did for me. I might be forgetting a lot of things but one thing that never fades inside me is good friendship. To all the girls of Groningen to whom I owe an apology for my intolerable character I have to say that "I'll remember you in my will."

Finishing off, I would like say a great thank you to my family for supporting me and for their understanding all these years. Your "silent presence" everywhere I go gives me the feeling I am never far from home. Thank you for your persistence, for your backing and for making me who I am, a man who believes in himself. Mother, your best wishes were not realized by chance but from you being there for me. Foteini, I am truly sorry for the way things ended. Your heart was like a lighthouse to me, giving meaning and reason to my life. Both good and bad times through this adventure shaped me into a better person and for this I am really grateful to you. You are always in my heart and mind.

After a stormy period of time, this thesis was successfully completed thanks to the concrete persistence and support of a very special person, Despoina Tsinalidou. Despoina, your positive thinking brought me to where I am now. Thank you and stay always like this.

Georgios K. Ouzounis
Alexandroupoli
December 8, 2008

Bibliography

- [1] B. Appleton and H. Talbot. Efficient path openings and closings. In *Mathematical Morphology: 40 Years On; Proc. 7th Int. Symp. Math. Morphology*, pages 33–42, Paris, 18-20 April 2005.
- [2] J. A. Bangham, R. Harvey, P. D. Ling, and R. V. Aldridge. Morphological scale-space preserving transforms in many dimensions. *J. Electr. Imag.*, 5:283–299, 1996.
- [3] J. A. Bangham, R. Harvey, P. D. Ling, and R. V. Aldridge. Nonlinear scale-space from n-dimensional sieves. In *Proceedings IEEE ECCV'96*, volume 1064 of *Lecture Notes in Computer Science*, pages 189–198, 1996.
- [4] N. Bouaynaya, M. Charif-Chefchaoui, and D. Schonfeld. Theoretical foundations of spatially-variant mathematical morphology part I: Binary images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):823–836, 2008.
- [5] N. Bouaynaya and D. Schonfeld. Theoretical foundations of spatially-variant mathematical morphology part II: Gray-level images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):837–850, 2008.
- [6] U.M. Braga-Neto and J. Goutsias. Multiresolution connectivity: An axiomatic approach. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 159–168, 2000.
- [7] U.M. Braga-Neto and J. Goutsias. Connectivity on complete lattices: New results. *Comp. Vis. Image Understand.*, 85:22–53, 2002.
- [8] U.M. Braga-Neto and J. Goutsias. A multiscale approach to connectivity. *Comp. Vis. Image Understand.*, 89:70–107, 2003.
- [9] U.M. Braga-Neto and J. Goutsias. A theoretical tour of connectivity in image processing and analysis. *J. Math. Imag. Vis.*, 19:5–31, 2003.
- [10] U.M. Braga-Neto and J. Goutsias. Grayscale level connectivity: Theory and applications. *IEEE Trans. Image Proc.*, 13(12):1567–1580, 2004.
- [11] E. J. Breen and R. Jones. Attribute openings, thinnings and granulometries. *Comp. Vis. Image Understand.*, 64(3):377–389, 1996.

- [12] J. M. H. Du Buf and M. M. Bayer, editors. *Automatic Diatom Identification*. Series in Machine Perception and Artificial Intelligence. World Scientific Publishing Co., Singapore, 2002.
- [13] V. Caselles and P. Monasse. Grain filters. *J. Math. Imag. Vis.*, 17:249–270, 2002.
- [14] F. Cheng and A. N. Venetsanopoulos. An adaptive morphological filter for image processing. *IEEE Trans. Image Proc.*, 1:533–539, 1992.
- [15] A.C.S. Chung. Image segmentation methods for detecting blood vessels in angiography. In *Int. Conf. Automation, Robotics and Computer Vision*, pages 1–6, 2006.
- [16] J. Crespo, R. W. Schafer, J. Serra, C. Gratin, and F. Meyer. The flat zone approach: a general low-level region merging segmentation method. *Signal Processing*, 62:37–60, 1997.
- [17] R. Diestel. *Graph Theory*. New York: Springer-Verlag, 1997.
- [18] Y. P. Du and D. L. Parker. Vessel enhancement filtering in three-dimensional MR angiograms using long-range signal correlation. *J. Magn. Reson. Imag.*, 7:447–450, 1997.
- [19] A. N. Evans. Vector area morphology for motion field smoothing and interpretation. *IEE Proc.-Vis. Image Signal Process.*, 150:219–226, 2003.
- [20] N. Flasque and M. Desvignes. Accurate detection of 3d tubular tree structures. In *Proc. Int. Conf. Image Processing*, pages Vol III: 436–439, 2000.
- [21] A. F. Frangi, W. J. Niessen, and M. A. Viergever K. L. Vincken. Multiscale vessel enhancement filtering. In W. M. Wells, A. Colchester, and S.L Delp, editors, *In Medical Image Computing and Computer-Assisted Intervention, MICCAI 1998, Proceedings*, volume 1496 of *Lecture Notes in Computer Science*, pages 130–137. Springer Verlag, Berlin, Germany, 1998.
- [22] A.F. Frangi, A.A. Amini, and E. Bullitt. Vascular imaging. *IEEE Trans. Medical Imaging*, 24(4):433–435, April 2005.
- [23] D. Gatica-Perez, C. Gu, M. T. Sun, and S. Ruiz-Correa. Extensive partition operators, gray-level connected operators, and region merging/classification segmentation algorithms: Theoretical links. *IEEE Trans. Image Proc.*, 10:1332–1345, 2001.
- [24] D. Gimenez and A. N. Evans. An evaluation of area morphology scale-spaces for colour images. *Comp. Vis. Image Understand.*, 110(1):32–42, 2008.
- [25] S. E. Grigorescu, N. Petkov, and P. Kruizinga. Comparison of texture features based on gabor filters. *IEEE Trans. Image Proc.*, 11(10):1160–1167, 2002.
- [26] H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, Boston, 1994.
- [27] H. J. A. M. Heijmans. Morphological filters. In *Proc. Summer School Morph. Image Signal Proc.*, Zakopane, Poland, 1995.
- [28] H. J. A. M. Heijmans. Composing morphological filters. *IEEE Trans. Image Proc.*, 6(5):713–723, 1997.
- [29] H. J. A. M. Heijmans. Connected morphological operators for binary images. *Comp. Vis. Image Understand.*, 73:99–120, 1999.

- [30] A. C. Jalba, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Morphological hat-transform scale spaces and their use in texture classification. In *Proc. Int. Conf. Image Proc. 2003*, volume I, pages 329–332, Barcelona, Spain, September 14-17 2003.
- [31] A. C. Jalba, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Morphological hat-transform scale spaces and their use in pattern classification. *Pattern Recognition*, 37(5):901–915, 2004.
- [32] A. C. Jalba, M. H. F. Wilkinson, and J. B. T. M. Roerdink. Automatic diatom identification using contour analysis by morphological curvature scale spaces. *Machine Vision and Applications*, 16(4):217–228, 2005.
- [33] A. C. Jalba, M. H. F. Wilkinson, and J. B. T. M. Roerdink. Shape representation and recognition through morphological curvature scale spaces. *IEEE Trans. Image Processing*, 15(2):331–341, 2006.
- [34] A. C. Jalba, M. H. F. Wilkinson, J. B. T. M. Roerdink, M. M. Bayer, and S. Juggins. Automatic diatom identification using contour analysis by morphological curvature scale spaces. IWI-report 2001-9-05, Institute for Mathematics and Computing Science, University of Groningen, Groningen, The Netherlands, 2001.
- [35] R. Jones. Connected filtering and segmentation using component trees. *Comp. Vis. Image Understand.*, 75:215–228, 1999.
- [36] J. C. Klein. *Conception et réalisation d'une unité logique pour l'analyse quantitative d'images*. PhD thesis, Nancy University, France, 1976.
- [37] T. Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Comp. Vision Graph. Image Proc.*, 48:357–393, 1989.
- [38] K. Krissian. Flux-based anisotropic diffusion applied to enhancement of 3-d angiogram. *IEEE Trans. Medical Imaging*, 21(11):1440–1442, November 2002.
- [39] R. Lerallut, E. Decencièrre, and F. Meyer. Image filtering using morphological amoebas. In *Mathematical Morphology: 40 Years On; Proc. 7th Int. Symp. Math. Morphology*, pages 13–22, Paris, 18-20 April 2005.
- [40] R. Lerallut, E. Decencièrre, and F. Meyer. Image filtering using morphological amoebas. *Image Vis. Comput.*, 25(4):395–404, 2007.
- [41] P. Maragos. Pattern spectrum and multiscale shape representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11:701–715, 1989.
- [42] P. Maragos and G. Evangelopoulos. Leveling cartoons, texture energy markers, and image decomposition. In *Proc. Int. Symp. Math. Morphology (ISMM) 2007*, pages 125–138, 2007.
- [43] P. Maragos and R. D. Ziff. Threshold superposition in morphological image analysis systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(5):498–504, 1990.
- [44] G. Matheron. *Random Sets and Integral Geometry*. John Wiley, 1975.
- [45] G. Matheron and J. Serra. Strong filters and connectivity. In J. Serra (Ed.), editor, *Image Analysis and Mathematical Morphology*, volume 2, pages 141–157. Academic Press: London, 1988.

- [46] A. Meijster, M. A. Westenberg, and M. H. F. Wilkinson. Interactive shape preserving filtering and visualization of volumetric data. In *Fourth IASTED Conf Comp. Signal Image Proc., SIP2002*, pages 640–643, Kauai, Hawaii, USA, August 12-14 2002.
- [47] A. Meijster and M. H. F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):484–494, 2002.
- [48] F. Meyer. Levelings, image simplification filters for segmentation. *J. Math. Imag. Vis.*, 20(1–2):59–72, 2004.
- [49] P. Monasse and F. Guichard. Fast computation of a contrast invariant image representation. *IEEE Trans. Image Proc.*, 9:860–872, 2000.
- [50] P. Monasse and F. Guichard. Scale-space from a level lines tree. *J. Vis. Commun. Image Repres.*, 11:224–236, 2000.
- [51] B. Naegel, N. Passat, N. Boch, and M. Kocher. Segmentation using vector-attribute filters: Methodology and application to dermatological imaging. In *Proc. Int. Symp. Math. Morphology (ISMM) 2007*, pages 239–250, 2007.
- [52] L. Najman and M. Couprie. Building the component tree in quasi-linear time. *IEEE Trans. Image Proc.*, 15:3531–3539, 2006.
- [53] M. Orkisz, M. Hernandez-Hoyos, P. Douek, and I. Magnin. Advances of blood vessel morphology analysis in 3D magnetic resonance images. *Mach. Vis. Graph.*, 9:463–471, 2000.
- [54] G. K. Ouzounis and M. H. F. Wilkinson. Countering oversegmentation in partitioning-based connectivities. In *Proc. Int. Conf. Image Processing.*, pages 844–847, 2005.
- [55] G. K. Ouzounis and M. H. F. Wilkinson. Second-order connected attribute filters using max-trees. In C. Ronse, L. Najman, and E. Decenciere, editors, *Mathematical Morphology: 40 Years On; Proc. 7th Int. Symp. Math. Morphology*, volume 30 of *Computational Imaging and Vision*, pages 65–74, Dordrecht, 2005. Springer-Verlag.
- [56] G. K. Ouzounis and M. H. F. Wilkinson. Filament enhancement by non-linear volumetric filtering using clustering-based connectivity. In Nanning Zheng, Xiaoyi Jiang, and Xuguang Lan, editors, *Advances in Machine Vision, Image Processing, and Pattern Analysis, International Workshop on Intelligent Computing in Pattern Analysis/Synthesis, IWICPAS*, volume 4153 of *Lecture Notes in Computer Science*, pages 317–327. Springer, 2006.
- [57] G. K. Ouzounis and M. H. F. Wilkinson. Mask-based second-generation connectivity and attribute filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):990–1004, 2007.
- [58] G. K. Ouzounis and M. H. F. Wilkinson. Hyperconnected attribute filters based on k-flat zones for 3d medical imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, Submitted.
- [59] G. K. Ouzounis and M. H. F. Wilkinson. Partition-induced connections and operators for pattern analysis. *Pattern Recognition*, Submitted.
- [60] N. Ray and S.T. Acton. Inclusion filters: a class of self-dual connected operators. *IEEE Trans. Image Proc.*, 14(11):1736–1746, 2005.
- [61] C. Ronse. Openings: Main properties, and how to construct them. Unpublished manuscript, Philips Research Laboratory Brussels, 1990.

- [62] C. Ronse. Set-theoretical algebraic approaches to connectivity in continuous or digital spaces. *Journal of Mathematical Imaging and Vision*, 8:41–58, 1998.
- [63] P. Salembier. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *IEEE Trans. Image Proc.*, 9(4):561–576, 2000.
- [64] P. Salembier, P. Brigger, J. R. Casas, and M. Pardàs. Morphological operators for image and video compression. *IEEE Trans. Image Proc.*, 5:881–898, 1996.
- [65] P. Salembier, A. Oliveras, and L. Garrido. Anti-extensive connected operators for image and sequence processing. *IEEE Trans. Image Proc.*, 7:555–570, 1998.
- [66] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Trans. Image Proc.*, 4:1153–1160, 1995.
- [67] J. Serra. *Image Analysis and Mathematical Morphology*, volume 1. Academic Press, New York, 1982.
- [68] J. Serra, editor. *Image Analysis and Mathematical Morphology. II: Theoretical Advances*. Academic Press, London, 1988.
- [69] J. Serra. Connectivity on complete lattices. *Journal of Mathematical Imaging and Vision*, 9:231–251, 1998.
- [70] J. Serra. Connections for sets and functions. In *Fundamenta Informaticae*, volume 41, pages 147–186, 2000.
- [71] J. Serra. Viscous lattices. In *Mathematical Morphology; Proc. 6th Int. Symp. Math. Morphology*, pages 79–90, 2002.
- [72] J. Serra. A lattice approach to image segmentation. *Journal of Mathematical Imaging and Vision*, 24:83–130, 2006.
- [73] J. Serra and P. Salembier. Connected operators and pyramids. In *Proc. of SPIE Image Algebra and Mathematical Morphology*, volume 2030, pages 65–76, San Diego, California, USA., 1993.
- [74] A. Sofou, C. Tzafestas, and P. Maragos. Segmentation of soilsection images using connected operators. In *Proc. Int. Conf. Image Proc. 2001 (ICIP-2001)*, pages 1087–1090, Thessaloniki, Greece, October 2001.
- [75] P. Soille. Beyond self-duality in morphological image analysis. *Image and Vision Computing*, 23:249–257, 2005.
- [76] P. Soille. Constrained connectivity for hierarchical image decomposition and simplification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(7):1132–1145, 2008.
- [77] P. Soille, E. Breen, and R. Jones. Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(5):562–567, 1996.
- [78] H. Talbot and B. Appleton. Efficient complete and incomplete path openings and closings. *Image Vis. Comput.*, 25(4):416–425, 2007.
- [79] R. E. Tarjan. Efficiency of a good but not linear set union algorithm. *J. ACM*, 22:215–225, 1975.
- [80] I. R. Terol-Villalobos and D. Vargas-Vázquez. Openings and closings with reconstruction criteria: a study of a class of lower and upper levelings. *J. Electron. Imaging*, 14(1):013006, 2005.

- [81] F. Tushabe and M. H. F. Wilkinson. Image preprocessing for compression: Attribute filtering. In *Proc. World Congress on Engineering & Computer Science 2007*, pages 999–1005, 2007.
- [82] F. Tushabe and M. H. F. Wilkinson. Content-based image retrieval using combined 2d attribute pattern spectra. In *Proc. Clef'2007*, 2008. In press.
- [83] C. Tzafestas and P. Maragos. Shape connectivity: Multiscale analysis and application to generalized granulometries. *Journal Math. Imaging and Vision*, 17:109–129, 2002.
- [84] E. R. Urbach, N. J. Boersma, and M. H. F. Wilkinson. Vector-attribute filters. In C. Ronse, L. Najman, and E. Decenciere, editors, *Mathematical Morphology: 40 Years On; Proc. 7th Int. Symp. Math. Morphology*, pages 95–104. Springer-Verlag, Dordrecht, 2005.
- [85] E. R. Urbach, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):272–285, 2007.
- [86] E. R. Urbach and M. H. F. Wilkinson. Shape-only granulometries and grey-scale shape filters. In H. Talbot and R. Beare, editors, *Mathematical Morphology; Proc. 6th Int. Symp. Math. Morphology*, pages 305–314, Collingwood, Australia, 2002. CSIRO Publishing.
- [87] L. Vincent. Morphological area openings and closings for greyscale images. In *Proc. NATO Shape in Picture Workshop*, pages 197–208, Driebergen, The Netherlands, September 1992.
- [88] L. Vincent. Morphological grayscale reconstruction in image analysis: application and efficient algorithm. *IEEE Trans. Image Proc.*, 2:176–201, 1993.
- [89] L. Vincent. Granulometries and opening trees. *Fundamenta Informaticae*, 41:57–90, 2000.
- [90] M. A. Westenberg, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Volumetric attribute filtering and interactive visualization using the max-tree representation. *IEEE Trans. Image Proc.*, 16(12):2943–2952, 2007.
- [91] M. H. F. Wilkinson. Attribute-space connected filters. In C. Ronse, L. Najman, and E. Decencie, editors, *Mathematical Morphology: 40 Years On; Proc. 7th Int. Symp. Math. Morphology*, pages 85–94. Springer-Verlag, Dordrecht, 2005.
- [92] M. H. F. Wilkinson. Attribute-space connectivity and connected filters. *Image Vis. Comput.*, 25:426–435, 2007.
- [93] M. H. F. Wilkinson, H. Gao, W. H. Hesselink, J. E. Jonker, and A. Meijster. Concurrent computation of attribute filters using shared memory parallel machines. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1800–1813, 2008.
- [94] M. H. F. Wilkinson and M. A. Westenberg. Shape preserving filament enhancement filtering. In W. J. Niessen and M. A. Viergever, editors, *Proc. MICCAI'2001*, volume 2208 of *Lecture Notes in Computer Science*, pages 770–777, 2001.
- [95] O. Wink, W. J. Niessen, and M. A. Viergever. Fast delineation and visualization of vessels in 3-D angiographic images. *IEEE Transactions on Medical Imaging*, 19:337–346, 2000.
- [96] W.C.K. Wong, A.C.S. Chung, and S.C.H. Yu. Local orientation smoothness prior for vascular segmentation of angiography. In *European Conf. Computer Vision*, pages Vol II: 353–365, 2004.

- [97] P.J. Yim, M. Kayton, W. Miller, S. Libutti, and P.L. Choyke. Automated detection of blood vessels using dynamic programming. *Pattern Recognition Letters*, 24(14):2471–2478, October 2003.
- [98] N. Young and A. N. Evans. Psychovisually tuned attribute operators for pre-processing digital video. *IEE Procs. Vision, Image and Signal Processing*, 150(5):277–286, 2003.

