TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Dogu Baran Aydogan

Contour Tree Connectivity and Analysis of Microstructures

Julkaisu 1223 • Publication 1223

Tampere 2014

Tampereen teknillinen yliopisto. Julkaisu 1223
Tampere University of Technology. Publication 1223

Dogu Baran Aydogan

Contour Tree Connectivity and Analysis of Microstructures

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Rakennustalo Building, Auditorium RG202, at Tampere University of Technology, on the 31st of July 2014, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2014

ISBN 978-952-15-3314-3 (printed)
ISBN 978-952-15-3331-0 (PDF)
ISSN 1459-2045

Abstract

The connectivity of microstructures is directly related to the physical properties of materials. Currently, the Euler number is the most popular measure of connectivity. It is an elegant topological invariant, however, it does not provide information about cavities or the proximities and sizes of objects. In this thesis, an alternative measure called contour tree connectivity (CTC) is developed and its applications for the analysis of microstructures are studied.

CTC is derived from contour trees that are used in the first publication to represent complex binary images with simple graphs. By analyzing contour trees, CTC produces new connectivity information that is not provided by other approaches described in the literature. Contour tree representation of binary images and CTC can be computed for any dimensions of data and topology as explained in the second publication. Moreover, CTC is designed to be a scalar between 0 and 1, which makes it easy to use and understand.

In this thesis, the use of CTC for analyzing microstructures is presented in two studies. In the first study, the microstructure of trabecular bone is analyzed in relation to its mechanical strength. In the second study, the relationship between microstructures and the fluid flow within materials are examined. The results from these studies show that CTC contributes to the understanding of how the structural properties of materials are linked to their physical properties.

To conclude, with its unique properties, CTC complements the structural information provided by currently used measures. This makes it an important image analysis tool for the study of the microstructures of materials such as soil, paper, filters and food products as well as biomaterials and biological tissues.

Acknowledgements

The work presented in this thesis was done during 2010-2014 at the Department of Electronics and Communications Engineering, Tampere, Finland.

I would like to express my gratitude to my supervisor, Prof. Jari Hyttinen, for his support throughout my research. I thank the reviewers, Prof. Peter Lee of The University of Manchester, UK and Dr. Silvia Biasotti of The National Research Council, Italy for their advice, careful examination and evaluation of my thesis. I thank Prof. Hannu Aro and Dr. Niko Moritz of Turku University Hospital and University of Turku for providing me an excellent dataset of trabecular bone images and helping me analyse it. I am deeply grateful to Jenny and Antti Wihuri Foundation for the financial support during 2010-2013, without which this work would not have been possible.

I am grateful to my colleagues and the staff from the former Ragnar Granit Institute and the Department of Biomedical Engineering for their support throughout all the painful transitions and for helping me to continue my research minimally affected. For that, I would particularly like to thank my supervisor Prof. Jari Hyttinen, Dr. Pasi Kauppinen and office secretary Soile Lönnqvist. I am thankful to the staff of Tampere University Hospital restaurant and FinnMedi Oy who prepared my lunches and cleaned my room everyday, without them I could not have functioned.

I sincerely thank my past and present colleagues from the image processing group, Tuukka Arola, Markus Hannula, Nathaniel Narra, Tomas Cervinka and Dr. Jarno Tanskanen for the invaluable discussions and all-around help with my research. I would like to thank my friends and colleagues Leena Jaatinen, Alper Cömert and Fikret Emre Kapucu for being the joy at work and for reminding me that my life is not only about the PhD. I warmly thank all my

other friends, especially my friends in Tampere from Turkey for sharing with me the challenges of living abroad and bringing the warmth of Turkey in the darkest and coldest days of Finland.

I cannot thank Mustafa Emre Karagözler enough for being the most faithful friend ever. Even though he has been on the other side of the world, he has always kept in touch, encouraged and inspired me. Without his help I could have never finished my PhD, nor he without mine.

I am deeply grateful to Elli's family, especially her parents Markku and Pirjo Käpylä for supporting me and being my family far away from home.

There are no words to describe how thankful I am to my family, my father Hasan, my mother Hüsniye and my sister Demet for their unending support throughout my life. No matter how far they are, they have always trusted in me and motivated me to do my best, which carried me to the end of my PhD. Lastly, I thank Elli for having the biggest loving heart. My life, work and this thesis would have been meaningless without you.

Dogu Baran Aydogan

Tampere, May 2014

List of publications

This thesis is written based on the following four selected publications of the author. The publications are referred to as [Publication-X] in the text, where X is the number of the publication. All publications are printed with permissions from the publishers.

Publication-I Aydogan D.B. & Hyttinen J. "Binary image representation by contour trees", Proceedings of SPIE 8314, Medical Imaging 2012: Image Processing, 83142X, 4-10.02.2012, San Diego, CA, USA

Publication-II Aydogan D.B. & Hyttinen J. "Contour tree connectivity of binary images from algebraic graph theory", IEEE International Conference on Image Processing (ICIP), 15-22.09.2013, Melbourne, Australia

Publication-III Aydogan D.B., Moritz N., Aro H. & Hyttinen J. "Analysis of trabecular bone microstructure using contour tree connectivity", Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013, volume 8150, Springer, 428-435, Eds: Mori, Kensaku, Sakuma, Ichiro, Sato, Yoshinobu, Barillot, Christian, and Navab, Nassir, 2013

Publication-IV Aydogan D.B. & Hyttinen J. "Characterization of microstructures using contour tree connectivity for fluid flow analysis", J R Soc Interface 2014 11: 20131042

Author's contributions

Author's contributions for each article are as follows:

Publication-I Author designed, implemented and tested the method that is used to represent the complex structure of trabecular bones using contour trees. Trabecular bone images were acquired by Niko Moritz, University of Turku, Finland. Other test images were created by the author.

Publication-II Author developed the contour tree connectivity (CTC) as a novel connectivity measure for binary images. Author generated the test images and calculated all the results presented in the work.

Publication-III Author computed the CTC values of the trabecular bones, compared ten morphometric parameters with ultimate bone strength and performed the classification tests. All the images, ultimate strengths of bones and morphometric parameters other than CTC were acquired by the second and third authors.

Publication-IV Author designed and created the test samples, estimated their permeabilities using modeling, calculated the connectivity parameters and did the statistical analysis described in the article.

All articles were completely written by the author. They were read, commented and approved by the co-authors.

CONTENTS

GLOSSARY

List of abbreviations

μCT	microcomputed tomography
2D	two dimensional
3D	three dimensional
BMD	bone mineral density ($\frac{g}{cm^3}$)
BV/TV	bone volume fraction
Conn.D	connectivity density ($1/mm$)
CT	computed tomography
CTC	contour tree connectivity
DA	degree of anisotropy
DNOM	nominal opening dimension (m)
DPLS	direct pore scale modeling
EDT	Euclidean distance transform
EN	Euler number
EP	Euler parameter
FEM	finite element method
LDA	linear discriminant analysis
PCP	parameter for connected pairs
PCT	pruned contour tree
SCT	supplemented contour tree
SIMPLE	semi implicit method for pressure linked equation
SMI	structure model index
Tb.N	trabecular number ($1/mm$)
Tb.Pf	trabecular bone pattern factor
Tb.Sp	trabecular separation (mm)
Tb.Th	mean trabecular thickness (mm)

List of symbols

β_k	k^{th} Betti number
χ	Euler number
$\circ$	morphologic opening operator
δ^+	up degree
δ^-	down degree
Γ	proportion of connected pairs
γ	index of a critical point
λ_n	$(n+1)^{th}$ smallest eigenvalue
$\mathbb{R}^n$	n dimensional real coordinate space
$\mathcal{B}_\epsilon$	ball of radius ϵ
$\mathcal{G}$	a graph
$\mathcal{O}$	Landau's symbol
$\overline{\lambda_1}$	normalized second smallest eigenvalue or contour tree connectivity
σ_U	ultimate tensile strength (Pa)
τ	connectivity function
Υ_t^+	upper level set at isovalue t
Υ_t^-	lower level set of at isovalue t
d_{nom}	nominal opening dimension (m)
f	a function
H	Hessian matrix
K	permeability (m^2)
M	a manifold
R^2	coefficient of determination
xor	exclusive or operator

1 INTRODUCTION

It is an important research problem to find out how the physical properties of materials and their microstructures are related. To achieve this, a variety of image analysis methods to study the structural properties of materials such as connectivity have been proposed. Connectivity is an important property for several reasons. For example, the flow and transport properties within porous objects are governed by the connectivity of the pore network [55], the stiffness of fibrous objects depends strongly on the number of connected/touching fibers [32] and the mechanical strength of the trabecular bone is related to the connectivity of the trabecular structure [139]. Despite efforts to link connectivity with various physical properties, there is, however, a lack of a good quantitative measure for it. The only scalar measure that is applicable for any geometry is the Euler number that is a topological invariant. Although there are a few other scalar measures of connectivity such as parameters based on percolation theory, their applications are not common and they cannot provide intuitive connectivity measures.

This thesis proposes a novel scalar measure for connectivity and introduces its uses in predicting the physical properties of materials from images of microstructures. In particular, it is targeted at obtaining a connectivity measure that is both effective and intuitive for complex microstructures.

1.1 Aims and contributions

The aims of the thesis are the following:

1. To develop a new interpretation for connectivity which has more geometrical relevance, rather than purely topological or functional.
2. To design and create effective algorithms to quantify connectivity with a scalar, that is easy to understand and use.
3. To investigate how this new geometrical interpretation of connectivity relates to real-life processes, such as flow and mechanical properties of materials.

The contributions of the thesis are the following:

1. A new connectivity measure that is named the contour tree connectivity (CTC) is developed. [Publication-I, Publication-II]
2. CTC is designed to be a scalar, that is between 0 and 1. [Publication-II]
3. Structural connectivity information quantified with CTC is shown to be linked to mechanical and fluid flow properties of microstructures. [Publication-III, Publication-IV]

1.2 Organization of the thesis

The thesis comprises six chapters. In Chapter 2, related work is summarized. This chapter starts by explaining the connectivity concept used in image processing. Following this, the common problems of shape analysis and how they are solved in computer vision applications are summarized. Lastly, image analysis and the connectivity of microstructures are explained.

In Chapter 1.1, the aim of the thesis and the main contributions are summarized. The methods developed to achieve the aims of the thesis are given in Chapter 3. The method chapter is the core of this thesis and provides extended explanations of the ideas presented in [Publication-I,Publication-II]. Here, mainly the new connectivity measure that we named *contour tree connectivity (CTC)* is described.

In order to develop CTC, we combine tools from computational topology and algebraic graph theory. From computational topology, *contour trees* are used. The traditional use of contour trees is to represent high-dimensional scalar data with simple graphs that are later combined with user interfaces for visualization purposes. From algebraic graph theory, we use the *algebraic connectivity* that is a standard tool to assess the connectivity properties of graphs. In Section 3.1, the definition of contour trees is given. Contour trees are commonly defined based on Morse theory. Following this tradition, we also start by giving a contour tree definition for Morse functions in subsection 3.1.2. Later we extend this definition to non-Morse functions in subsection 3.1.3. Following the definitions, we give detailed pseudo codes of the algorithms for each step that is used to compute contour trees, in subsection 3.1.4. Since we do not provide the actual codes in this thesis, we intended to give all the information needed to replicate our method in the pseudo codes. In Sections 3.2 and 3.3, we explain how binary images are represented using contour trees and how CTC is calculated from this representation.

Chapter 4 is divided into three main sections, each of which explains an application of CTC. In Section 4.1, the results from [Publication-II] and the performance of CTC are studied. In this section, several binary 2D and 3D images are analyzed to show the different connectivity interpretation of CTC and to summarize the fundamental geometric changes that alter CTC. In Sections 4.2 and 4.3, the focus is on the analysis of microstructures using CTC. These sections give the results from [Publication-III] and[Publication-IV]. Section 4.2 explains an application of CTC where we characterize the microstructures of trabecular bone biopsies. In this section, we show that connectivity information quantified by CTC improves the prediction of ultimate bone strength. In Section 4.3, fluid flow in microstructures is studied using artificial samples and computer models. It is shown that CTC, unlike other popular connectivity measures, is capable of discriminating different microstructures.

Chapters 5 and 6 are the discussions and conclusion chapters where we elaborate on our methodological approach in the development of CTC, the experimental and performance results, the drawbacks of CTC and how we intend to improve CTC in our future work.

2 RELATED WORK

In this chapter, we provide a review of the literature that is related to our work. We start with explaining the basic understanding of the connectivity concept and its applications in image processing. In the second section, we give an overview of the shape analysis methods and applications used in general for computer vision. In the third section, we focus in the image analysis of microstructures and the methods used to study connectivity.

2.1 Connectivity in image processing

Connectivity is a fundamental concept in image processing since it sets up the spatial relationship between the elements of an image. For example, pixel connectivity defines in which way pixels in 2D or 3D relate to each other [95]. This establishes the base for the definition of connected components as well as the morphological operators.

As an alternative to classical pixel-based filtering in image processing, morphological operators are extended to gray-scale images which leads to connected operators and filters [14, 133]. As an extension, connectivity is interpreted as a spatial property that generates flat regions and how the regions are nested within each other [98, 134]. This interpretation aims to fulfill the need to define objects and process them separately which has become necessary with the increase in image and video data. For example, a related representation called the *tree of shapes* has been proposed in [74], which mainly aims to structure the inclusion relationship of shapes by their saturation, $sat(.)$. The $sat(.)$ operator fills the holes of a level set in order to delineate shapes.

Another interpretation of connectivity is given in [123] which presents an alternative to the traditional crisp connectivity interpretation in image processing. Here images are considered as fuzzy in nature due to imaging and other factors. For example, objects in images are heterogeneous instead of flat regions due to resolution limits and noise, or the boundaries in images are not sharp but blurry. This fuzzy connectivity interpretation aims to treat such uncertainties as realistically as possible by means of a function called the *affinity*. Affinity determines the strength of connectedness among the elements by how spatially close they are and how close their intensities are. For segmentation applications, affinity is used to explain how objects *hang together*; it is also used to define *fuzzy connected components*.

2.2 Overview of shape analysis methods and applications

Shape analysis in general focuses on how to describe geometric properties of object(s). After images are acquired, it is customary to start by segmentation which refers to marking of the objects that need to be studied. By segmentation, the information in an image is reduced only to a certain number of objects, which makes rest of the image redundant. As a result, a new image is created, where objects of interest or shapes are represented with 1s and the rest with 0s. Such images are called *binary images*. It is worth mentioning that, by shape, it is referred to an appearance of an object which is independent of location, scale and rotational effects [39].

Shape analysis has an abundant number of applications in addition to merely studying structural properties of objects. Information acquired from shapes are used for recognition [105], registration [99], matching [53, 126], retrieval [116] and segmentation [49]. Due to the wide spectrum of shape analysis applications, there have been a significant amount of research done to describe and represent shapes in both 2D and 3D images [64, 116, 137]. Shape description refers to the extraction of useful numeric descriptors of shapes. 2D shape descriptors are commonly categorized into two as contour and region based methods [137]. For example perimeter is a contour descriptor and area is a region descriptor. Unlike 2D, 3D shapes are mostly represented with meshes; thus most 2D descriptors cannot be extended to 3D. Due to these differences, 3D shape descriptors are studied separately from 2D counterparts. In [116], 3D shape descriptors are classified into

feature based, geometry based and graph based. With feature based methods, geometrical features of shapes such as volume, convex hull and moments are referred. Geometry based approaches utilize similarities between shapes depending on different views or point sets. Graph based methods utilize meaningful graph representation of shapes such as skeletons [113], Morse-Smale complexes [109], contour trees [22] or Reeb graphs [10, 91]. Graph representation of shapes have also a fundamental place in computational topology and used commonly for removing topological features for simplifying high dimensional scalar fields and visualization [9, 143].

With the availability of imaging equipments, it has become common to have a collection of images and shapes to be used for certain tasks. Shapes belonging to a certain population are studied using tools of statistical shape analysis in which the main aim is to find variability of shapes. Similarities among shapes are initially studied using correspondences between landmarks [39]. The correspondences are used for example to align shapes with respect to location, scale or rotation, this is also referred as procrustes analysis [12]. Establishing correspondences in a set of shapes creates shape spaces by which distances between shapes can be computed. This makes it possible to compute mean shapes, shape models and variability of shapes within populations, commonly by using principal component analysis (PCA) [49]. Starting from late 1990's there has been a major interest in developing effective feature descriptors which assign each point of a shape a multi-dimensional feature vector that describes local and geometrical properties at that point. Particularly the use of eigenvalues and eigenvectors of Laplace-Beltrami operator initiated a whole set of feature descriptors which are generally called as spectral shape descriptors [19, 29]. Notable spectral shape descriptors include ShapeDNA [93], global point signature [96], heat kernel signature [112] and wave kernel signature [6].

2.3 Image analysis of microstructures

Figure 2.1 shows example images of segmented structures of objects acquired from a microcomputed tomography (μCT) device. The complex geometry of many materials at microscale introduces a new set of problems for shape analysis. Contrary to the applications of connectivity in image processing and shapes in computer vision, when studying microstructures, the main goal is to analyze and extract useful structural information that relates to physical attributes of materials.

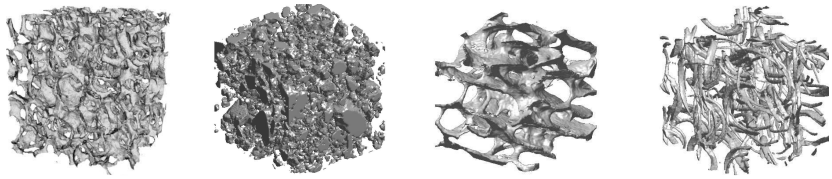


Figure 2.1: Example images obtained using a μCT device. (a) Low density ceramic material for insulation and filtering applications (courtesy of Prof. Erkki Levänen, Department of Materials Science, TUT, Tampere), (b) soil (courtesy of Markus Hannula, ELT, TUT, Tampere and BioMediTech, Tampere), (c) trabecular bone, (d) fibrous biomaterial scaffold (courtesy of Prof. Minna Kellomäki, ELT, TUT, Tampere and BioMediTech, Tampere)

Common characteristics for microstructures that can be computed for all geometries are the Euler number, volume, surface area, the integral of mean curvature and their densities [82]. However, due to the variability of shapes and applications, there have been specific solutions proposed to analyze particular microstructures. For example, foreground and background of foam, fibrous, sintered and powdered structures provide different information about the physical properties of materials. Flow and transport properties within porous objects are governed by the pore network (background information) and the stiffness of fibrous objects depends strongly on the number of touching fibers (foreground information) [32]. Traditionally *mean intercept length* (MIL) is used to describe mean size of constituents and anisotropy of microstructures of fibrous materials [47]. Briefly, when computing MIL, the length of intersections between a set of parallel lines and the image is computed. By repeating this procedure for several different angles, a measure for the orientation of the structure is determined. This technique is commonly used also for determining the anisotropy of the trabecular bone [79]. *Structure model index* (SMI) is an other measure to characterize trabecular bones, which quantifies the type of trabecular structure [50]. SMI basically computes and compares the change in the surface area in the case of a small increase in volume. For plate-like structures $SMI = 0$ and for rod-like structures $SMI = 3$. For granular structures, granulometry is used to characterize structures by means of generating profiles for the size and shape distributions of objects that form images [15, 68]. Granulometry mainly applies the morphological opening operator iteratively on an image with increasing sizes for the structure element. The changes in each opening operation, such as the sum of pixel values, are associated with the size of objects [122, 124, 127].

2.4 Connectivity of microstructures

Among several characterizations, understanding the role of connectivity has become a very active research area since early 2000's. A paradigm shift towards studying connectivity has been pointed out in the fields of hydrogeology, surface hydrology, geomorphology, landscape ecology and soil physics [92]. For understanding flow in microstructures, connectivity is an important property to study because it complements the information from local spatial variances and higher order correlations, which are traditionally used to model heterogeneity. There are several application and structure specific approaches proposed to study connectivity in the literature. For example, for foam materials, watershedding is generally used to partition pore space [67, 106]. This is followed by estimating pore size distributions and strut thicknesses which is used as a connectivity measure for the pore network [5, 60]. Another common approach is to use the medial axis transform and skeletonization to partition pore space and form pore networks for foam materials to analyze connectivity [61, 77, 78].

A major problem associated with connectivity is the lack of a unique mathematical definition. Being an intuitive notion, the quantification of connectivity is based on how it is interpreted. Application specific interpretations put aside, it is common to interpret connectivity as either a static or functional (dynamic) property [92]. Static interpretations regard connectivity only as a structural property whereas functional connectivity measures involve physical processes which can be estimated experimentally or with modeling such as the effective hydraulic conductivity [57].

The differences in the interpretation of connectivity creates a problem when it comes to comparing connectivities of different structures. For example since there are no common interpretations for foam, sintered or fibrous materials, it is challenging to make comparisons. Therefore general methods, such as the *connectivity function* (τ), that are applicable for any geometry are needed. Connectivity function is defined as the probability that two points belong to the same connected component [3]:

$$\tau(x, y) = P(x \longleftrightarrow y | x, y \in C) \tag{2.1}$$

where x, y are two elements of the image, P is probability and C is a connected component.

In order to simplify the connectivity function, it is common to compute τ as a function of a shift, α, for which $y = x + \alpha$ is assumed [121]. Although connectivity function can be computed for any geometry, it does not provide a scalar measure. It is desirable to study connectivity using a scalar since multi dimensional comparisons are challenging. However, such measures are extremely limited. To our knowledge, Euler number (χ) is the only scalar connectivity measure that can be computed for any geometry. In addition to Euler number, we encountered two other related scalar measures. These are the proportion of connected pairs (Γ) [51] and the nominal opening dimension (d_{nom}) [128]. Although these parameters are connectivity related, their applications are not as general as Euler number's. Also they do not provide intuitive connectivity information. However since they are used in the literature as scalar connectivity measures to study flow [92], we next explain them along with Euler number.

2.4.1 Euler number (χ)

The most commonly used connectivity measure for binary images is the Euler number (χ), which is described with the Euler-Poincaré formula as the alternating sum of Betti numbers, β_k , $\chi = \sum_k (-1)^k \beta_k$. For 2D, Betti numbers are the number of connected components (β_0) and the number of holes (β_1). For 3D, they are the number of connected components (β_0), the number of tunnels (β_1) and the number of voids (β_2). χ is easily calculated using graph counts as follows [59]:

$$\chi = v - e + f - s \tag{2.2}$$

where v is the number of vertices, e is the number of edges, f is the number of faces and s is the number of solids.

The connectivity parameter named the *connectivity density* (Conn.D) is perhaps the best example for the use of Euler number. Conn.D is commonly used to study trabecular bone microstructure [80]. For a 3D binary image that is composed of a single connected component with no voids, $1 - \chi$ gives the number of tunnels that exist in the connected component. The tunnels in the case of trabecular bone are mostly associated with trabeculae. This makes Conn.D, which is defined as the number of tunnels per unit volume, a very useful tool to study trabecular bone structure.

The density of Euler number is an important measure when studying foam materials as well. The topological properties of foam are similar to random networks consisting of nodes and edges. Since density of connections can be computed using Euler number, it is also possible to estimate the density of nodes [70]. This approach is also used to compute the number of touching fibers in non-woven materials [82].

Euler number is very easy to understand which makes it an elegant topological invariant. On the other hand, for most of the practical purposes, topology alone is not sufficient to describe connectivity properties and other geometrical information related to connectivity becomes important. For example, it is indicated in [83] that, to provide bone ingress and angiogenesis, the thickness of interconnections between pores of a biomaterial foam should be at least 100 μm. Such information regarding geometrical dimensions are not provided by Euler number. Although Euler number includes information regarding the number of connected components or holes, it does not provide information about the proximity of connected components, sizes of holes or cavities.

In order to include more information considering the local variations in structures, Euler number is commonly computed on level or excursion sets obtained by thresholding scalar fields [104, 128, 129]. An example application to a series of thresholded images along with corresponding χ values are shown in Fig.2.2. By repeating the experiments for several threshold values, functions for χ are obtained. An example is shown in Fig.2.3a that is applied on the sample shown in Fig.2.2.

2.4.2 Proportion of connected pairs (Γ)

Proportion of connected pairs is a parameter used in percolation theory which studies the simplest phase transition with nontrivial critical behavior [18]. For example, assume that each cell of a 3D grid has a probability of p to be occupied and $1-p$ to be empty. Suppose that p is increased from 0 to 1, after some certain p value, there starts to exist a cluster of occupied cells that connects the top and the bottom of the grid, such a cluster is said to *percolate*. One of the most important problems in percolation theory is to investigate the percolation threshold, that is the value of p after which a percolating cluster starts to occur. In general, percolation theory studies occupied cells, clusters and their properties with the changes in p [111]. Among the many properties of clusters, we found that proportion of connected pairs is used as a scalar connectivity measure [51].

High resolution images of many microstructures have sharp boundaries, which makes it straightforward to segment unoccupied or permeable (pores) and occupied or impervious (material) regions using simple techniques such as thresholding [101]. The proportion of connected pairs is defined as:

$$\Gamma = (1/n^2) \sum_{i=1}^{N} n_i^2 \tag{2.3}$$

where n is the total number of permeable cells, N is the number of connected permeable regions and n_i is the number of permeable cells in the i^{th} region.

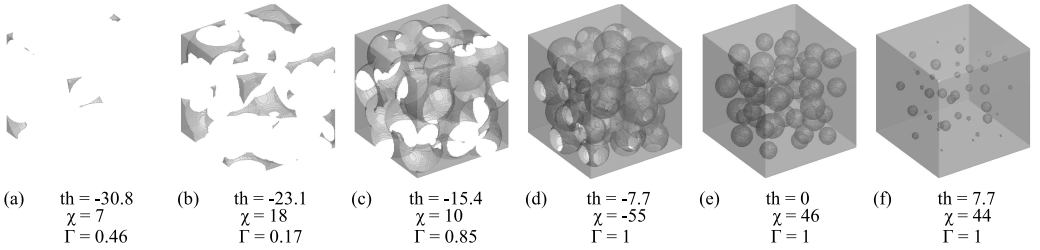


Figure 2.2: Euler number (χ) and proportion of connected pairs (Γ) corresponding to the excursion sets obtained by thresholding distance transform of an artificial 3D image. From left to right the value of the threshold (th) is increased. At $th = 0$, the resultant image is the image of the permeable region. (Reproduced from [Publication-IV].)

Γ shows the ratio of the pairs of cells that are connected to all the pairs of permeable cells. One common case in which Γ does not provide information is when there is a medium with a single fully connected permeable region, for which $\Gamma = 1$. In order to include the effects of local variability, similar to χ, Γ is also computed iteratively on the level sets [92]. Fig.2.2 and Fig.2.3b shows an example to this approach.

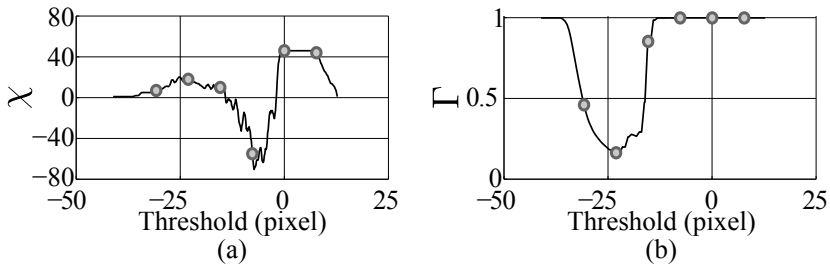


Figure 2.3: (a) Euler number (χ) and (b) proportion of connected pairs (Γ) as a function of threshold. The functions are obtained using the sample in Fig.2.2. Dots on the graphs indicate the location of the χ and Γ values calculated in Fig.2.2. (Reproduced from [Publication-IV].)

2.4.3 Nominal opening dimension (d_{nom})

One of the most common techniques to estimate pore (opening) size distribution and nominal opening size, d_{nom}, is to use the morphological opening operator which eliminates features smaller than the structure element. By iteratively using the opening operator with increasing sizes of spheres as structure elements, the opening size distribution is obtained [128, 129].

Let $I \circ S_d$ denote the opening of image I with a spherical structure element with diameter d, S_d. Then, the cumulative opening size distribution of image I is a function of d that is defined as:

$$O(d) = \sum \text{xor}(I, I \circ S_d) \tag{2.4}$$

where xor denotes the *exclusive or* operator. The opening size distribution is $o(d) = O'(d)$. Using $o(d)$, d_{nom} is calculated by the following expected value:

$$d_{nom}(I) = \frac{\sum o(d).d}{\sum o(d)} \tag{2.5}$$

d_{nom} is also used as an estimate for the characteristic length scale [2]. The use of opening operator iteratively has also been presented in [5, 75] as a measure for the connectivity of porous biodegradable materials.

3 METHODS

This chapter explains the theoretical foundations and algorithms to implement the contour tree connectivity (CTC). The chapter is composed of three sections. First section starts with the description of contour trees and continues with the formal definition for Morse fields. Here, the definition given by Carr is taken as the main reference [22]. Generally, this definition is extended to be generalized for smooth functions using perturbation. Instead of this approach, contour trees are generalized using up- and down-degrees of isocomponents in section 3.1.3. To our knowledge, this the first contour tree definition for general smooth functions which does not rely on perturbation. In the last part of the first section, all the algorithms used to compute contour trees are given in detail. The algorithms are inspired from Carr's work in [22]. The main contribution of the thesis is given in the last two sections of the chapter, where binary image representation by contour trees and contour tree connectivity are explained.

3.1 Contour trees

3.1.1 Description of the contour tree

Contour trees describe topological properties of images by showing the evolution of contours as they appear, join, split or disappear by means of a graph. In order to explain the main idea behind contour trees, an example is given in Fig.3.1.

Assume that the data shown in Fig.3.1a is an elevation map. It is generally easier to visualize and understand the spatial changes of functions or a 2D image in this example

with manifolds as shown in Fig.3.1b. Because 3D visualization tools has not been always available, it has become customary to use contour plots as shown in Fig.3.1c. Notice that for each local maxima and minima a new class of contours that are nested each other appears. Additionally where saddles exist, the nested contour classes meet each other.

Contour trees depict this nesting and meeting relationship of contour classes with a graph. In Fig.3.1d the contour tree is overlaid on the input image. Local maxima and minima of the image are represented with red and blue vertices in the graph respectively, saddles are shown with green vertices. Fig.3.1e shows the same graph with the corresponding elevations of the vertices.

It is the simplicity of contour trees that make it an attractive tool for several purposes. The example demonstrates that without the need for 3D visualization or contour plots, important points of a function can be shown with simple graphs.

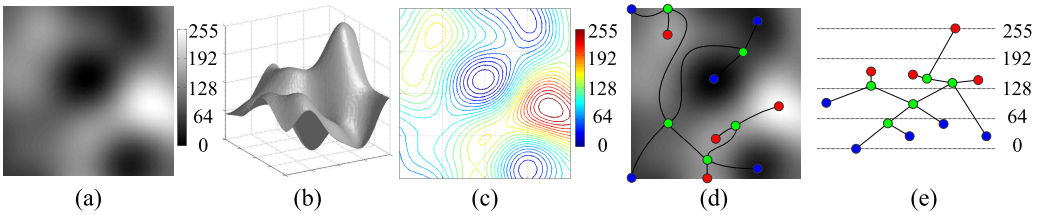


Figure 3.1: (a) Example elevation data (b) 3D visualization of the elevation (c) Contour plot of the elevation (d) Contour tree overlaid on the input image. Maxima, minima and saddles are shown with red, blue and green vertices respectively (e) Contour tree with vertices shown with corresponding elevations.

Contour trees are originally used for topography applications [13] and in many areas of geo-science involving the use of terrain profiles [30, 45]. Early applications of contour representation of terrains are used to describe features like peaks, pits, ridges and ravines [58, 114].

Contour trees are commonly used for abstraction and visualization of scalar fields [7, 8, 63]. They are also used to compute Reeb graphs to develop alternative abstractions [37, 120]. Contour trees are used together with user interfaces to select interesting isovalues, particularly when dimensions increase and visualization becomes challenging [16, 21, 23, 56]. Also they are used to design transfer functions for volume rendering [115, 130, 140]. In [85], the edges in contour trees are augmented with Betti numbers which gives additional topological information for the levelsets. Other visualization applications of contour trees

include development of web-based tools [100] and visualization of time-varying scalar fields [110].

It has become popular in the recent years to use contour trees in order to analyze scalar fields on top of abstraction and visualization applications. In [119], by comparing subtrees in contour trees, symmetric or repeating patterns in scalar fields are identified. Applications to find out similarities and differences between scalar fields are also developed [102, 103, 138].

Contour trees are also used in pattern recognition and machine vision. In [36], a method using contour representation is proposed to recognize company logos. In [38], topological information extracted from contour trees are used to identify key features in images which are used for example to identify craters on the surface of Mars and to locate breast tumors from optical tomography data.

3.1.2 Contour tree definition for Morse fields

A formal definition for contour trees can be given using the Morse theory [22]. Simply, Morse theory connects topology of a manifold with the function defined on it.

Definition 1 (Manifold)**.** *A topological space M is an n-dimensional manifold or an n-manifold if for all the points in M there exists an open ball that is homeomorphic to $\mathbb{R}^n$.*

Definition 2 (Critical point)**.** *Let M be an n-manifold and $f\colon M \to \mathbb{R}$ be a smooth mapping. A point $p \in M$ is called a critical point if all the partial derivatives of f at p are 0 with respect to the local coordinates of M, $(x_1, \cdots, x_n)$, i.e:*

$$\frac{\partial f}{\partial x_1}(p) = 0, \frac{\partial f}{\partial x_2}(p) = 0, \cdots, \frac{\partial f}{\partial x_n}(p) = 0, \tag{3.1}$$

Critical points of a function $f(x_1, \cdots, x_n)$ are categorized using its Hessian matrix, H_f, which is given as follows:

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{3.2}$$

Definition 3 (Nondegenerate and degenerate critical points)**.** *Let p be a critical point of function f. p is a nondegenerate critical point if $detH_f(p) \neq 0$, otherwise it is degenerate.*

Fig.3.2 shows various degenerate critical point examples for 2 and 3-manifolds. Notice that degeneracies can happen at points, curves or surface patches for 2-manifolds, in addition to these for 3-manifolds it can also happen on volume regions. The categorization of critical points helps to define Morse functions.

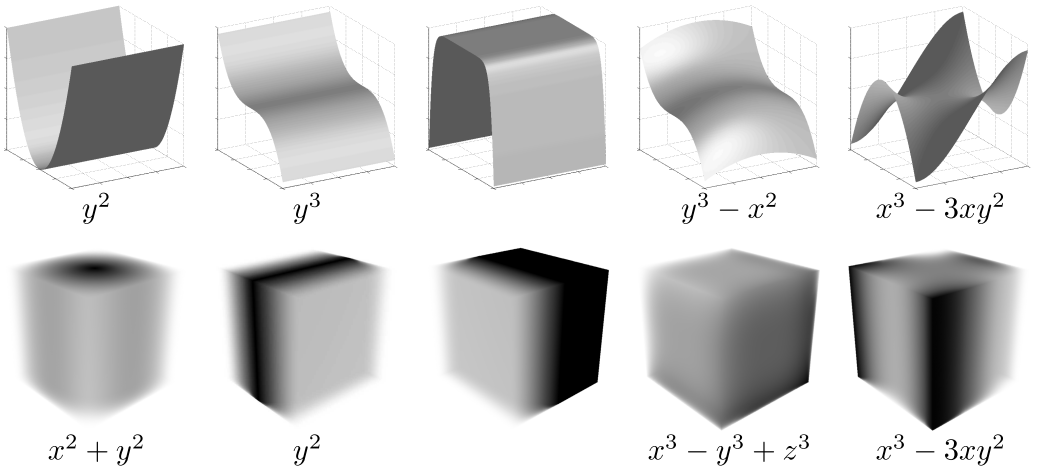


Figure 3.2: Upper and lower rows show example degenerate critical points for 2 and 3-manifolds. For 2-manifolds, on first and second columns, degeneracies occur along curves. Forth and fifth columns show degeneracies that occur at a single point whereas on third column, degeneracy occurs on a surface patch. A degeneracy on a surface patch of a 3-manifold is shown in the second column. First and fifth columns show degeneracies along curves for 3-manifolds. A volume region degeneracy is shown in the third column for 3-manifolds. In the forth column a degeneracy that occurs at a single point is shown.

Definition 4 (Morse function)**.** *A mapping $f\colon M \to \mathbb{R}$ is a Morse function if all its critical points are nondegenerate.*

Definition 5 (Index of a critical point)**.** *Let $f\colon M \to \mathbb{R}$ be a Morse function. The index $\gamma(p)$ of a critical point $p \in M$ is the number of the negative eigenvalues of $H_f(p)$.*

Morse lemma states that, it is possible to express a Morse function f about a critical point with a second order representation of the form $f(x_1, \cdots, x_n) = f(p) - \sum_{i=1}^{\gamma(p)} x_i^2 + \sum_{i=\gamma(p)+1}^{n} x_i^2$ [71]. It follows from Morse lemma that, nondegenerate critical points are isolated. Therefore, Morse functions have finitely many critical points.

Notice that for a 2D Morse function $f(x, y)$, the Hessian is $H_f = [\frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y \partial x}; \frac{\partial^2 f}{\partial x \partial y} \frac{\partial^2 f}{\partial y^2}]$. Possible indices for critical points are 0, 1 and 2. And according to Morse lemma $f(x, y) =$

$f(x_c, y_c) \pm x^2 \pm y^2$ around a critical point $\{x_c, y_c\}$. All three possible cases for f around a critical point are shown in Fig 3.3. Since it is rare to find the visualizations of the critical points on 3-manifolds, these are also given in Fig 3.3.

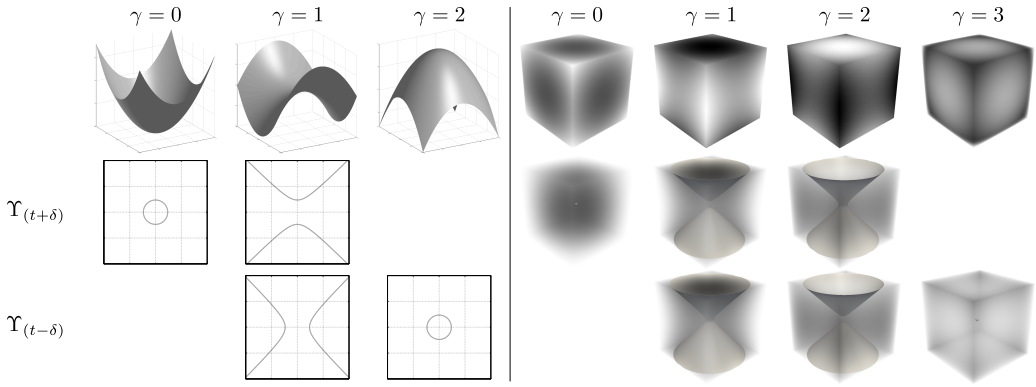


Figure 3.3: First row shows all possible cases for a critical point, p, of a Morse function f on 2 (left) and 3-manifolds (right). Indices of the critical points are shown on top. Second and third rows show the level sets $\Upsilon_{(t+\delta)}$ and $\Upsilon_{(t-\delta)}$ respectively for $t = f(p)$ and δ being a very small positive number.

Using a Morse function $f\colon M \to \mathbb{R}$, it is possible to learn about the topology of M from the level sets or contours of f.

Definition 6 (Level set, upper and lower level sets)**.** *The level set of a function $f\colon M \to \mathbb{R}$ at $t \in \mathbb{R}$ is the set $\Upsilon_t(f) = \{p \in M | f(p) = t\}$. Lower and upper levels sets are $\Upsilon_t^-(f) = \{p \in M | f(p) \leq t\}$ and $\Upsilon_t^+(f) = \{p \in M | f(p) \geq t\}$ respectively.*

t is generally called as *isovalue* in the literature. Notice that $\Upsilon_t(f)$ is the boundary of the upper and lower levels sets and $\Upsilon_t(f) = \Upsilon_t^-(f) \cap \Upsilon_t^+(f)$.

Definition 7 (Isocomponent)**.** *An isocomponent of a function $f\colon M \to \mathbb{R}$ at isovalue t is a single component of $\Upsilon_t(f)$.*

Definition 8 (Contour)**.** *Let M be an n-manifold and $f\colon M \to \mathbb{R}$ be any function. Then a contour of f is an isocomponent that is an isolated point or homeomorphic to any m-dimensional sphere where $m < n$.*

It is common in literature to use synonyms for contours, such as *isolines* or *isocontours* for 2-manifolds and *isosurfaces* for 3-manifolds. In this text, the term contour is used in general for any dimension.

Corollary 1. *If a function is Morse then all of its isocomponents are contours. Non-Morse functions may or may not have isocomponents which are not contours.*

Corollary 2. *Each point of a Morse function belongs to exactly one contour.*

Fig.3.3 shows the contours of $\Upsilon_{(t-\delta)}$ and $\Upsilon_{(t+\delta)}$ around the critical points. Notice that if δ is picked to be a very small positive number, for a minimum or a maximum, a single contour is approaching to the critical points whereas for a saddle two contours are coinciding. Based on the examples and observations made for 2 and 3-manifolds, it is possible to identify critical points by the number of contours in their neighborhood using the following definitions.

Let M be an n-manifold ($n \geq 2$) and $f\colon M \to \mathbb{R}$ be a Morse function. Assume that $f(p) = t$ for a point $p \in M$. Let $\mathcal{B}_\epsilon(p)$ be the ball of radius ϵ centered at p and $f|\mathcal{B}_\epsilon(p)$ denote the restriction of f to $\mathcal{B}_\epsilon(p)$. Lastly let δ and ϵ be very small positive numbers.

Definition 9 (Up- and down-degree of a point)**.** *The up-degree, $\delta^+(p)$, of f at p is the number of contours of f at isovalue $t+\delta$ that intersects $\mathcal{B}_\epsilon(p)$, as $\delta, \epsilon \to 0$. The number of contours of at isovalue $t-\delta$ is called the down-degree, $\delta^-(p)$.*

Using δ^+ and δ^- definitions, regular points, maxima, minima and saddles of a Morse function $f\colon M \to \mathbb{R}$ can be identified as follows. A point $p \in M$ is a:

- regular if $\delta^+(p) = 1$ and $\delta^-(p) = 1$.
- local maximum if $\delta^+(p) = 0$.
- local minimum if $\delta^-(p) = 0$.
- saddle either if
  - $\delta^+(p) = 2$
  - $\delta^-(p) = 2$

Notice the difference that for n-manifolds where $n > 2$, different contours of $f|\mathcal{B}_\epsilon(p)$ may belong to the same contour of f. Therefore, a regular point which has $\delta^+(p) = \delta^-(p) = 1$ can be a critical point according to Definition 2. Due to this difference the critical points defined using up- and down-degrees are separated with the following definition.

Definition 10 (Component critical point)**.** *A component critical point is a point p where both $\delta^+(p)$ and $\delta^-(p)$ are not equal to 1.*

For the rest of the text unless stated otherwise, with regular points, points with $\delta^+ = \delta^- = 1$ are referred. The conditions of $\delta^+(p) = 2$ or $\delta^-(p) = 2$ for saddle cases are also given special names in the literature.

Definition 11 (Join and split)**.** *A join is a saddle where* $\delta^+(p) \geq 2$*. A split is saddle where* $\delta^-(p) \geq 2$*.*

The inequalities in Definition 11 will be useful in the next section when degeneracies are handled.

In order to explain how critical points of a function give information about the topology of a manifold, changes in Betti numbers are illustrated in Fig.3.4 for the smooth function $f\colon M \to \mathbb{R}$ used in example Fig.3.1a. A step by step investigation of the changes in the lower level sets $\Upsilon_t^-(f)$ while $t \in \mathbb{R}$ is increased from $-\infty$ to ∞ are listed below. The values of the indices of the critical points, up-degrees and down-degrees are given in parenthesis.

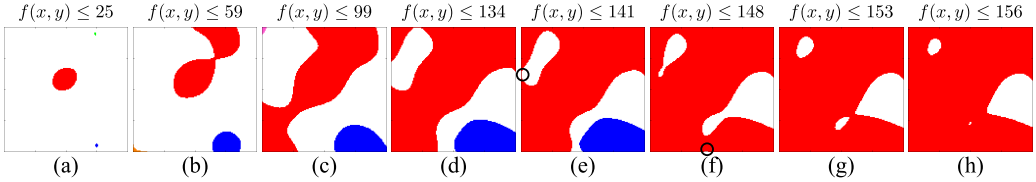


Figure 3.4: Topological changes in $\Upsilon_t^-(f)$ for several values of t which are given on top. Different colors (red, blue) indicate different components. Merging components at a saddle point are shown with the same color (red). The circle in (e) shows a point on the boundary which causes a topological change although it is not a component critical point. The circle in (f) shows a local maxima which is on the boundary thus does not alter β_1.

- Between $(-\infty, t_{gmin})$ where t_{gmin} is the global minimum of f, $\Upsilon_t^-(f) = \{\}$, no topology change occurs.
- $(\gamma = 0, \delta^- = 0, \delta^+ = 1)$ At each local minimum a new component is introduced. β_0 increases by 1.
- At each saddle one of the following happens:
  - $(\gamma = 1, \delta^- = 2, \delta^+ = 1 \text{ or } 2)$ Two components merge. β_0 decreases by 1.
  - $(\gamma = 1, \delta^- = 1, \delta^+ = 2)$ A hole forms inside a component. β_1 increases by 1.
- $(\gamma = 2, \delta^- = 1, \delta^+ = 0)$ At each local maximum a hole is filled. β_1 decreases by 1. (For manifolds with boundary, maxima at boundaries do not alter topology.)

- Between (t_{gmax}, ∞) where t_{gmax} is the global maximum of f, $\Upsilon_t^-(f) = M$, no topology change occurs.

Same investigation can be done using the upper level sets which yields a similar result. It is evident that the observations made above are valid for all Morse functions on surfaces such as for 2D images. However, for higher dimensions there will be more varieties for the saddle cases.

For the sake of completeness, an elegant outcome of the Morse theory is given without proof in Equation 3.3. Curious readers are recommended to read [71, 143] for the proof of this expression and further information on Morse theory.

$$\chi(M) = \sum_{i=0}^{n} (-1)^i \beta_i = \sum_{i=0}^{n} (-1)^i (\#\gamma_i) \tag{3.3}$$

where M is a closed manifold and $\#\gamma_i$ is the number of critical points with index i.

In image processing, 2 and 3-manifolds are traditionally not closed but compact with boundary and critical points can be at the boundaries as shown in Fig.3.4e-f. In short, Equation 3.3 is not useful in its purest form. Regardless of this, the earlier demonstration shows that encountering a component critical point during the transitions of the isovalues alter the topology of level sets. Therefore, an equivalence relation for contours can be defined as follows:

Definition 12 (Equivalent contours)**.** *Let c and c' be two contours of a Morse function $f\colon M \to \mathbb{R}$ at isovalues t and t' respectively where $t < t'$. And let M_Γ be the manifold $M_\Gamma = \Upsilon_t^+ \cap \Upsilon_{t'}^-$. Then c are c' are equivalent if there exists a path Γ in M_Γ that connects a point of c with a point of c' such that there is no point of Γ which belongs to a contour of any component critical point of f.*

It follows from Definition 12 that all points in Γ belong to contours that are equivalent to c and c'. Therefore, the transition of the isovalue $t \to t'$ spans an infinite number of contours that are all equivalent. Contours that are equivalent are also said to belong to the same *contour class*.

The theorem that is given next uses the contour class concept and establishes the base for the contour tree definition.

Theorem 1. *Let $f\colon M \to \mathbb{R}$ be a Morse function. Then the contours for all component critical points of f belong to a contour class with a single contour and the contours for all regular points of f belong to a contour class with infinite contours.*

Proof. Since all critical points of a Morse function are isolated and each point of f belongs to a single contour by Corollary 2, a contour that includes a component critical point is not equivalent to any other contour and is therefore the sole member of its contour class. Second part of the theorem can be proved by showing that no contour on a regular point is the only member of its contour class. Let $p \in M$ be a regular point with $f(p) = t$ and c be the contour on p. Then $\delta^+(p) = \delta^-(p) = 1$, thus there exists contours $c^+ \subset \Upsilon_{(t+\delta)}(f)$ and $c^- \subset \Upsilon_{(t-\delta)}(f)$ which are in the same contour class with c. □

The correspondence between single member contour classes for contours on component critical points and infinite member contour classes for contours on regular points is used in [20] to give a graph theoretical definition of the contour tree as follows:

Definition 13 (Contour tree for Morse functions)**.** *The contour tree, $\mathcal{G}(V, E)$, of a Morse function $f\colon M \to \mathbb{R}$ is a graph with a vertex set V and edge set E such that:*

1. *For each minima, maxima, join and split of f, there is a vertex $v_i \in V$.*
2. *For each contour at v_i, through its transition to the next component critical point at v_j, there is an edge $(v_i, v_j) \in E$.*

3.1.3 Generalized contour trees for smooth functions

In the previous section main properties of Morse functions are summarized and the formal definition of the contour tree is given based on Morse theory. Unfortunately, the functions in most practical applications are not Morse and have degenerate critical points. Some examples of degeneracies were given earlier in Fig.3.2.

In Fig.3.5, level sets $\Upsilon_{(t-\delta)}$ and $\Upsilon_{(t+\delta)}$ are given for the degeneracies shown in Fig.3.2. On the second column for 2-manifolds and forth column of both manifolds, $\delta^+ = \delta^- = 1$. Based on the discussion made for Definition 9, these degeneracies can be discarded from the set of critical points and are categorized as regular.

Notice that nondegenerate critical points are always isolated, however all isolated critical points are not nondegenerate. An example to this is shown with $x^3 - 3xy^2$ in Fig.3.2. Therefore degeneracies happen both on isolated points and on a set of topologically connected points.

All the points of a Morse function are identified earlier by using the degree definitions given in Definition 9. In order to follow a similar path for a non-Morse function, degree definitions can be generalized to isocomponents. Let M_D be any isocomponent of a

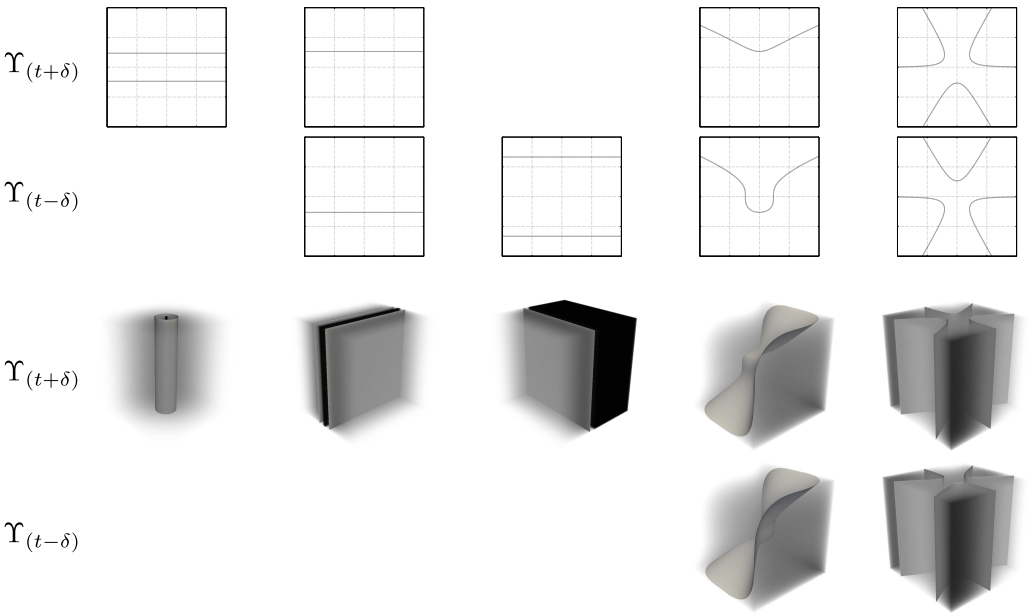


Figure 3.5: First two rows show the level sets $\Upsilon_{(t+\delta)}$ and $\Upsilon_{(t-\delta)}$ for the degenerate critical points of 2-manifolds given in Fig.3.2. Last two rows show the same level sets for the degenerate critical points of 3-manifolds given in Fig.3.2. $\delta^+ = \delta^- = 1$ for the 2-manifold in the second column also for both manifolds of the forth column. Degeneracy happens on a single point for 2-manifolds on the forth and fifth columns. A degenerate volume is shown for a 3-manifold in the third column.

smooth function $f\colon M \to \mathbb{R}$ at isovalue t. Since f can be non-Morse, M_D is not necessarily a contour (Corollary 1). Let ∂M_D be the boundary of M_D. Notice that $\forall p \in M_D \setminus \partial M_D$, $\delta^+(p) = \delta^-(p) = 0$ and $\forall p \in \partial M_D$ there exists contours of $\Upsilon_{(t-\delta)}$ and/or $\Upsilon_{(t+\delta)}$. Based on this observation, the degree definitions for points can be extended to any isocomponent.

Let $f\colon M \to \mathbb{R}$ be any smooth function and $\mathcal{B}_\epsilon(p)$ be the ball of radius ϵ centered at $p \in M$ where ϵ is a very small positive number. Let M_D be an isocomponent of f at isovalue t.

Definition 14 (Up- and down-degree of an isocomponent)**.** *The up-degree of M_D, $\delta^+(M_D)$, is the number of distinct contours of f that intersects $\mathcal{B}_\epsilon(p)$ of all $p \in M_D$ at isovalue $t+\delta$, as $\delta, \epsilon \to 0$. The number of distinct contours at isovalue $t-\delta$ is called the down-degree, $\delta^-(M_D)$.*

The up- and down-degree symbols for both points and isocomponents are same since the context makes it clear which one is on concern.

The points of any smooth function $f\colon M \to \mathbb{R}$ can be identified using the degrees of isocomponents. Let M_D be any isocomponent of f then all points $p \in M_D$ are:

- regular if $\delta^+(M_D) = 1$ and $\delta^-(M_D) = 1$.
- local maxima if $\delta^+(M_D) = 0$ and $\delta^-(M_D) \geq 1$.
- local minima if $\delta^-(M_D) = 0$ and $\delta^+(M_D) \geq 1$.
- saddles either if
  - $\delta^+(M_D) \geq 2$ and $\delta^-(M_D) \neq 0$
  - $\delta^-(M_D) \geq 2$ and $\delta^+(M_D) \neq 0$

Definition 15 (Regular and critical isocomponents)**.** *A regular isocomponent is an isocomponent of a function with $\delta^+ = \delta^- = 1$, otherwise it is a critical isocomponent.*

If f is Morse, then the identification of the points are exactly the same as the one made earlier using Definition 9. Notice that there are additional conditions to identify critical isocomponents. This is because, with the inclusion of degenerate points, it is possible to have a join or a split at a local extrema. Therefore, the condition of $\delta^- = 0$ or $\delta^+ = 0$ is added in the saddle identification in order to separate local extrema.

With Definition 15 it is possible to generalize contour classes to *isocomponent classes*. For this, the equivalence relation of contours given in Definition 12 is extended for isocomponents of smooth functions.

Definition 16 (Equivalent isocomponents)**.** *Let c and c' be two isocomponents of a smooth function $f\colon M \to \mathbb{R}$ at isovalues t and t' respectively where $t < t'$. And let M_Γ be the manifold $M_\Gamma = \Upsilon_t^+ \cap \Upsilon_{t'}^-$. Then c are c' are equivalent if there exists a path Γ in M_Γ that connects a point of c with a point of c' such that there is no point of Γ which belongs to any critical isocomponent of f.*

Similarly, Theorem 1 can also be generalized for any smooth function.

Theorem 2. *Let $f\colon M \to \mathbb{R}$ be a smooth function. Then all the critical isocomponents of f belong to an isocomponent class with a single isocomponent and all the regular isocomponents of f belong to isocomponent classes with infinite isocomponents.*

Proof. By the definition of a critical isocomponent, $\delta^+ \neq \delta^- \neq 1$, therefore there is no other isocomponent that belongs to the same class. Second part of the theorem can be proved by showing that no regular isocomponent is the only member of its isocomponent class. Let $c \in M$ be a regular isocomponent at isolevel t. Then by definition $\delta^+(c) = \delta^-(c) = 1$, thus there exists contours $c^+ \subset \Upsilon_{(t+\delta)}(f)$ and $c^- \subset \Upsilon_{(t-\delta)}(f)$ which are in the same isocomponent class with c. □

Finally, the contour tree of a smooth function can be defined as follows:

Definition 17 (Contour tree)**.** *The contour tree, $\mathcal{G}(V, E)$, of a smooth function $f\colon M \to \mathbb{R}$ is a graph with a vertex set V and edge set E such that:*

1. *For each critical isocomponent of f, there is a vertex $v_i \in V$.*
2. *For each isocomponent at v_i, through its transition to the next critical isocomponent at v_j, there is an edge $(v_i, v_j) \in E$.*

The main difference between the contour tree definition for Morse functions and smooth functions is that, vertices of the contour tree of a Morse function corresponds to a single point of the manifold whereas vertices of the contour tree of a random smooth function may contain a set of points that belong to an isocomponent of the manifold.

3.1.4 Contour tree computation algorithm

For the early applications of contour trees to simplify terrain profiles, a manual method is proposed in [13]. The first systematic computation approach to compute contour trees is given in [33, 125]. This approach is based on the observation that component critical points cause topological changes during the transition of the upper and lower level sets. One easy way to locate component critical points is to keep a record of the topological changes of the components during the transition of isovalues as demonstrated in subsection 3.1.2. During the sweep of the lower level sets, new components appear where there are local minima ($\delta^- = 0$) and connected components merge where there are splits ($\delta^- \geq 2$). However, it is clear that by sweeping the lower level sets and keeping a record of the connected components, one cannot detect where holes form and close, simply because during these topological changes, connected components are preserved. On the other hand, it can be observed from the example shown in Fig.3.4 that holes form where there are joins ($\delta^+ \geq 2$) and they close where there are local maxima ($\delta^+ = 0$). Therefore, sweeping

the upper level sets complements the sweep of the lower level sets. As a result, two sweeps together are able to locate all the component critical points. However, the algorithm in [125] requires that no more than two components merge at a component critical point and the boundaries are needed to be processed separately. The computational complexity of this algorithm is $\mathcal{O}(n\log n)$ for 2D where n is the number of vertices in the mesh, for 3D it is $\mathcal{O}(n^2)$. This algorithm is extended and improved in [117] to have a computational complexity of $\mathcal{O}(n\log n)$ for 3D as well.

One reason behind contour trees' popularity is the elegant algorithm given in [22] which is $\mathcal{O}(N + n\log n + t\alpha(t))$ where N is the number of cells, t is the number of component critical points and α is the inverse of the Ackermann function [21]. With this algorithm, the limitations of [125] are overcome, such as the separate handling of multisaddles and boundaries. Similar to [125], the computation of contour trees in this algorithm is done based on a sweep and merge approach.

In the sweep stage, up- and down-degrees of all points are computed. As a result, two graphs called the *join and split trees* are formed. In the merge step, these two trees are combined to form the contour tree. The algorithm given in [22] is easy to understand, fast and very well documented [20]. Due to these reasons, the contour tree computation method used in this thesis is based on this algorithm. Although the main framework is same as [22], in this work, different than simplicial meshes, the computation is done on the regular rectilinear grid. Also, the union-find data structure is not implemented in the code.

The computation method implemented in this thesis has similarities to [72] in which a region based contour tree computation algorithm is proposed. This algorithm is also based on [22], with the difference that it works on the rectilinear grid similar to this work.

There are several other algorithms proposed to compute contour trees other than the approaches based on sweep and merge. In [52], an erosion based idea is proposed. This algorithm starts with the complete mesh and iteratively removes the vertices, edges, faces and tetrahedra while keeping the topological connectivity between the critical points. It is an efficient algorithm with a computation complexity of $\mathcal{O}(n)$; however no proof was provided to show that the resultant structure is the contour tree. In [114] an algorithm to construct contour trees for 2D data is given. Here firstly the extrema and the saddles of the data are found by analyzing the stars of the simplices and later the extrema are connected

saddles appropriately. In [26, 27], spatial adjacency of pixels are used to compute Voronoi diagrams which are later used to generate contour trees.

There have also been several other algorithms proposed using the sweep and merge approach, particularly, based on the algorithm given in [22]. For example in [28], an algorithm which is inspired from [22] is given where initially all the component critical points are identified and in a later step these are connected appropriately. The computational complexity is given as $\mathcal{O}(N + t\log t)$. Also in [25], the method in [22] is extended to arbitrary interpolants. Another sweep and merge approach is given in [85] where Betti numbers are augmented on contour trees, this study is followed by [86] where a parallelizable algorithm is developed which is $\mathcal{O}(n + t\alpha(n))$.

The complete process used to compute contour trees in this thesis is shown in Fig.3.6. Below each process, the number of the corresponding algorithm is given which are explained in the coming subsections. For each block, inputs and outputs are indicated on the associated arrows. Vertices of trees are shown with a superscript $^{\cdot}$ and the edges with a superscript $^{\rightarrow}$. The data structures where the indices are stored are indicated with the subscript $_{index}$.

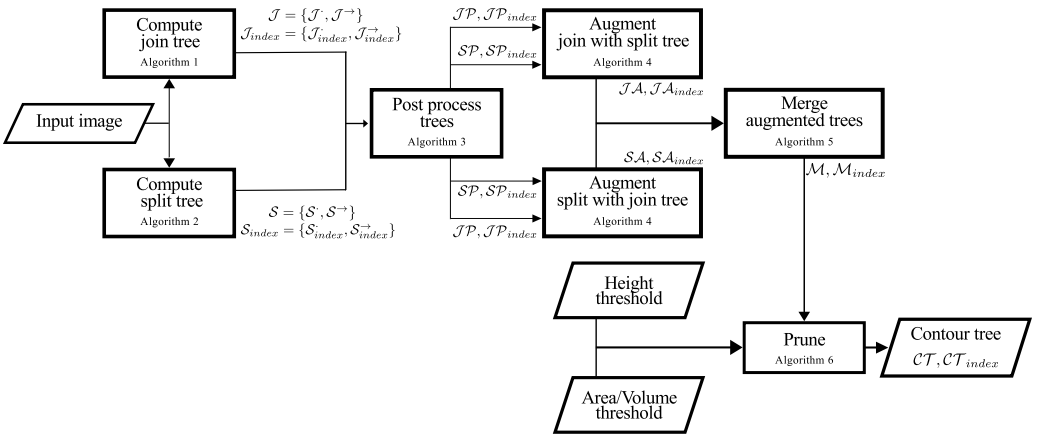


Figure 3.6: The complete process of the contour tree computation. Given an input image, first join and split trees are computed. The trees are post processed and augmented which prepare the graphs for merging. Given two pruning parameters (height and area/volume thresholds) the merged tree is cleaned which is the contour tree of the image.

Computations of join, $\mathcal{J} = \{\mathcal{J}^{\cdot}, \mathcal{J}^{\rightarrow}\}$, and split trees, $\mathcal{S} = \{\mathcal{S}^{\cdot}, \mathcal{S}^{\rightarrow}\}$, constitute the sweep stage. Merging starts with post processing and continues with the augmentation of the trees. Post processing and augmentation are done to prepare the graphs for the final merging operation. After the augmented trees are merged, a raw form of the contour tree is obtained which is referred to as the merge tree, $\mathcal{M} = \{\mathcal{M}^{\cdot}, \mathcal{M}^{\rightarrow}\}$,in the rest of the text. The algorithms used until this point are based on [22]. As commonly done in the literature, also in this work, merging is followed with a compulsory pruning step whose output is the contour tree, $\mathcal{CT} = \{\mathcal{CT}^{\cdot}, \mathcal{CT}^{\rightarrow}\}$,of the image. In the coming subsections, the building blocks of Fig.3.6 are explained with detailed pseudo codes of the underlying algorithms.

3.1.4.1 Pixel connectivity

Before giving the algorithms to compute the join and split trees, it has to be pointed out that in order to process the information using computers, images are digitized by means of sampling the data onto a rectilinear grid. These samples are the building blocks of digital images and are called *pixel*s. In 3D, pixels are also called *voxel*s. Since continuity is lost with sampling, a convention for connectivity of pixels is needed. Classical definitions for pixel connectivity is given in [95]. Next, the pixel connectivity convention used in his thesis is described.

In this work, it is assumed that each pixel is located in the middle of the *cell*s which are elements of the rectilinear grid. The cells are square shaped for 2D and cube for 3D images. The corners of each cell are called *pointel*s. Pointels are connected with *edgel*s. For 3D, each side of a cell is called a *facel*. Based on this framework, pixels can be connected with a variety of ways. For 2D images, 4-connectivity means that pixels are connected through the edgels but not through pointels; 8-connectivity means that pixels are connected both through edgels and pointels. Similar connectivity relations are set for 3D pixels as well. 6-connectivity in 3D implies that pixels are connected only through the facels. 18-connectivity means that pixels are connected through facels and edgels. In 26-connectivity facels, edgels and pointels connect pixels. These pixel connectivity arrangements are shown in Fig.3.7. The convention for pixel connectivity in this work is 8- connectivity for 2D and 26-connectivity for 3D.

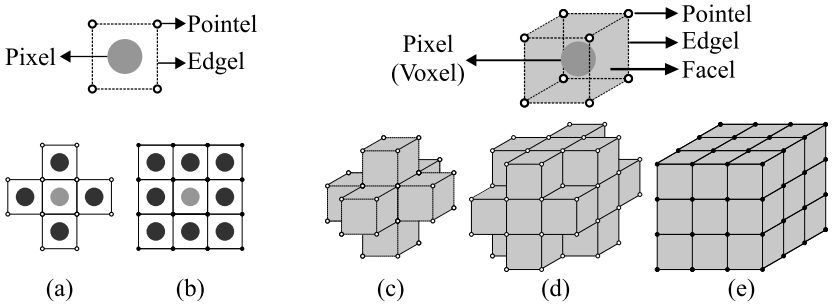


Figure 3.7: Pixel connectivity for 2D and 3D. Upper row shows the elements that make up the rectilinear grids for 2D and 3D images. (a) 4-connectivity (b) 8-connectivity (c) 6-connectivity (d) 6-connectivity (e) 26-connectivity

3.1.4.2 Computation of join and split trees

During the join sweep, a record of the connected components are kept throughout the transition of the isovalues from ∞ to $-\infty$. In the split sweep, isovalues are varied in the opposite direction. In order to discard the isovalues that are not present among the pixels, as the first step of the sweep stage, all pixels are sorted with respect to their intensities. Then they are analyzed in order. If a pixel is not connected to a connected component ($\delta^+ = 0$ or $\delta^- = 0$), it is a local/global extrema. If it is connected to multiple connected components ($\delta^+ \geq 2$ or $\delta^- \geq 2$), then it is a saddle. If it is connected to a single connected component ($\delta^+ = 1$ or $\delta^- = 1$), it is either a regular point of the join or split tree depending on the type of sweep. The iterations of the pixels continue until the last pixel in the queue is processed. The algorithms to compute the join tree and split tree are described below in Algorithm 1 and Algorithm 2 respectively. Notice that the outputs of the Algorithms 1 and 2 give the vertices of join and split trees $(\mathcal{J}^{\cdot}, \mathcal{S}^{\cdot})$ and the directed edges $(\mathcal{J}^{\rightarrow}, \mathcal{S}^{\rightarrow})$. Also the indices of the pixels that are on the vertices and edges are given in the output with $\mathcal{J}_{index}$ and $\mathcal{S}_{index}$. The indices are used to identify common vertices in the post processing also in the last step during pruning.

Algorithm 1: Computing the join tree

Input : Image I
Output: The join tree $\mathcal{J} = \{\mathcal{J}^{\cdot}, \mathcal{J}^{\rightarrow}\}$ and indices $\mathcal{J}_{index} = \{\mathcal{J}^{\cdot}_{index}, \mathcal{J}^{\rightarrow}_{index}\}$

Set $\mathcal{J}^{\cdot} = \{\}$, $\mathcal{J}^{\rightarrow} = \{\}$, $\mathcal{J}^{\cdot}_{index} = \{\}$, $\mathcal{J}^{\rightarrow}_{index} = \{\}$;
Sort the elements of I in descending order and form the list I_d;
Initialize an empty label image, L;
Set $max_L = 0$;
for $i = I_d(first) \rightarrow I_d(last)$ **do**
    Form the list $N_{indices}$, that has the indices of neighbors of i;
    Set $P_{indices} = \{\}$, $P_{labels} = \{\}$;
    for $n = N_{indices}(first) \rightarrow N_{indices}(last)$ **do**
        Get $L(n)$, that is the label of n from L;
        if $L(n) \neq 0$ & $L(n) \notin P_{labels}$ **then**
            Add n to $P_{indices}$, add $L(n)$ to P_{labels};
        end
    end
    if $P_{labels} = \{\}$ **then** // $\delta^+ = 0$
        Set $max_L = max_L + 1$;
        Set $\mathcal{J}^{\cdot}(max_L) = max_L$, $\mathcal{J}^{\rightarrow}(max_L) = 0$;
        Set $\mathcal{J}^{\cdot}_{index}(max_L) = i$, $\mathcal{J}^{\rightarrow}_{index}(max_L) = \{\}$;
        Set $L(i) = max_L$;
    else if P_{labels} *has 1 element* **then** // $\delta^+ = 1$
        Add i to $\mathcal{J}^{\rightarrow}_{index}(P_{labels}(1))$;
        Set $L(i) = P_{labels}(1)$;
    else if P_{labels} *has 2 or more elements* **then** // $\delta^+ \geq 2$
        Set $max_L = max_L + 1$;
        Set $\mathcal{J}^{\cdot}(max_L) = max_L$, $\mathcal{J}^{\rightarrow}(max_L) = 0$;
        Set $\mathcal{J}^{\cdot}_{index}(max_L) = i$, $\mathcal{J}^{\rightarrow}_{index}(max_L) = \{\}$;
        for $label = P_{labels}(first) \rightarrow P_{labels}(last)$ **do**
            $\mathcal{J}^{\rightarrow}(label) = max_L$;
            Replace $label$ in L with max_L;
        end
    end
end
$\mathcal{J}^{\rightarrow}(end) = max_L + 1$;

Algorithm 2: Computing the split tree

Input : Image I
Output: The split tree $\mathcal{S} = \{\mathcal{S}^{\cdot}, \mathcal{S}^{\rightarrow}\}$ and indices $\mathcal{S}_{index} = \{\mathcal{S}^{\cdot}_{index}, \mathcal{S}^{\rightarrow}_{index}\}$

Set $\mathcal{S}^{\cdot} = \{\}$, $\mathcal{S}^{\rightarrow} = \{\}$, $\mathcal{S}^{\cdot}_{index} = \{\}$, $\mathcal{S}^{\rightarrow}_{index} = \{\}$;
Sort the elements of I in descending order and form the list I_a;
Initialize an empty label image, L;
Set $max_L = 0$;
for $i = I_a(first) \rightarrow I_a(last)$ **do**
    Form the list $N_{indices}$, that has the indices of neighbors of i;
    Set $P_{indices} = \{\}$, $P_{labels} = \{\}$;
    for $n = N_{indices}(first) \rightarrow N_{indices}(last)$ **do**
        Get $L(n)$, that is the label of n from L;
        if $L(n) \neq 0$ & $L(n) \notin P_{labels}$ **then**
            Add n to $P_{indices}$, add $L(n)$ to P_{labels};
        end
    end
    if $P_{labels} = \{\}$ **then** // $\delta^- = 0$
        Set $max_L = max_L + 1$;
        Set $\mathcal{S}^{\cdot}(max_L) = max_L$, $\mathcal{S}^{\rightarrow}(max_L) = 0$;
        Set $\mathcal{S}^{\cdot}_{index}(max_L) = i$, $\mathcal{S}^{\rightarrow}_{index}(max_L) = \{\}$;
        Set $L(i) = max_L$;
    else if P_{labels} *has 1 element* **then** // $\delta^- = 1$
        Add i to $\mathcal{S}^{\rightarrow}_{index}(P_{labels}(1))$;
        Set $L(i) = P_{labels}(1)$;
    else if P_{labels} *has 2 or more elements* **then** // $\delta^- \geq 2$
        Set $max_L = max_L + 1$;
        Set $\mathcal{S}^{\cdot}(max_L) = max_L$, $\mathcal{S}^{\rightarrow}(max_L) = 0$;
        Set $\mathcal{S}^{\cdot}_{index}(max_L) = i$, $\mathcal{S}^{\rightarrow}_{index}(max_L) = \{\}$;
        for $label = P_{labels}(first) \rightarrow P_{labels}(last)$ **do**
            $\mathcal{S}^{\rightarrow}(label) = max_L$;
            Replace $label$ in L with max_L;
        end
    end
end
$\mathcal{S}^{\rightarrow}(end) = max_L + 1$;

3.1.4.3 Post processing and augmentation of join and split trees

Both outputs of the Algorithms 1 and 2 give a sorted array for the list of vertices starting from number 1. In the post processing step, the vertices of the join and split trees are given unique numbers. Additionally, if there are any pixels that have common indices in both trees, these are assigned the same vertex number. The algorithm used to post process the join and split trees is given in Algorithm 3. The last vertex of a post processed tree is a global extrema which does not have a directed edge to any other vertex in the

graph. Therefore, the last line of the algorithm sets this last edge to be directed to vertex 0 to indicate this.

An elegant algorithm to merge the join and split trees is given in [21]. This algorithm requires an augmentation step in which the vertices of the split tree are placed in the join and the vertices of the join tree are placed in the split tree. The augmentation algorithm is given in Algorithm 4. Here, simply the indices of the vertices of a tree are searched in the branches of the other tree. The vertices are not searched since common vertices are already handled in the post processing step.

Algorithm 3: Computing the post-processed join and split trees

Input : $\mathcal{J}$, $\mathcal{J}_{index}$, $\mathcal{S}$, $\mathcal{S}_{index}$
Output: $\mathcal{JP}\{\mathcal{JP}^{\cdot}, \mathcal{JP}^{\rightarrow}\}$, $\mathcal{JP}_{index}\{\mathcal{JP}^{\cdot}_{index}, \mathcal{JP}^{\rightarrow}_{index}\}$, $\mathcal{SP}\{\mathcal{SP}^{\cdot}, \mathcal{SP}^{\rightarrow}\}$, $\mathcal{SP}_{index}\{\mathcal{SP}^{\cdot}_{index}, \mathcal{SP}^{\rightarrow}_{index}\}$

Set $\mathcal{JP}^{\cdot}_{index} = \mathcal{J}^{\cdot}_{index}$, $\mathcal{JP}^{\rightarrow}_{index} = \mathcal{J}^{\rightarrow}_{index}$;
Set $\mathcal{SP}^{\cdot}_{index} = \mathcal{S}^{\cdot}_{index}$, $\mathcal{SP}^{\rightarrow}_{index} = \mathcal{S}^{\rightarrow}_{index}$;
Set $\mathcal{JP}^{\cdot} = \mathcal{J}^{\cdot}$, $\mathcal{JP}^{\rightarrow} = \mathcal{J}^{\rightarrow}$;
Set $\mathcal{SP}^{\cdot} = \mathcal{S}^{\cdot} + \mathcal{J}^{\cdot}(end)$, $\mathcal{SP}^{\rightarrow} = \mathcal{S}^{\rightarrow} + \mathcal{J}^{\cdot}(end)$;
Initialize an empty 2 column matrix, *comm*, for common vertices;
for $i = \mathcal{S}^{\cdot}_{index}(first) \rightarrow \mathcal{S}^{\cdot}_{index}(last)$ **do**
  for $j = \mathcal{J}^{\cdot}_{index}(first) \rightarrow \mathcal{J}^{\cdot}_{index}(last)$ **do**
    if $\mathcal{S}^{\cdot}_{index}(i) = \mathcal{J}^{\cdot}_{index}(j)$ **then** append $\{\mathcal{SP}^{\cdot}(i), \mathcal{JP}^{\cdot}(j)\}$ to *comm* ;
  end
end
for $i = comm(first) \rightarrow comm(last)$ **do**
  for $k = \mathcal{SP}^{\cdot}(first) \rightarrow \mathcal{SP}^{\cdot}(last)$ **do**
    if $\mathcal{SP}^{\cdot}(k) = comm(i, 1)$ **then** $\mathcal{SP}^{\cdot}(k) = comm(i, 2)$;
    if $\mathcal{SP}^{\rightarrow}(k) = comm(i, 1)$ **then** $\mathcal{SP}^{\rightarrow}(k) = comm(i, 2)$;
  end
end
Form $\mathcal{SP}_{unique}$ which has the unique vertices of $\mathcal{SP}^{\cdot}$ in ascending order;
Initialize $tmp\mathcal{SP}^{\cdot} = \mathcal{SP}^{\cdot}$, $tmp\mathcal{SP}^{\rightarrow} = \mathcal{SP}^{\rightarrow}$;
for $i = \mathcal{SP}_{unique}(first) \rightarrow \mathcal{SP}_{unique}(last)$ **do**
  if $\mathcal{SP}_{unique}(i) \notin comm(., 2)$ **then**
    for $k = \mathcal{SP}^{\cdot}(first) \rightarrow \mathcal{SP}^{\cdot}(last)$ **do**
      if $\mathcal{SP}^{\cdot}(k) = \mathcal{SP}_{unique}(i)$ **then** $tmp\mathcal{SP}^{\cdot}(k) = i + \mathcal{J}^{\cdot}(end)$;
      if $\mathcal{SP}^{\rightarrow}(k) = \mathcal{SP}_{unique}(i)$ **then** $tmp\mathcal{SP}^{\rightarrow}(k) = i + \mathcal{J}^{\cdot}(end)$;
    end
end
$\mathcal{SP}^{\cdot} = tmp\mathcal{SP}^{\cdot}$;
$\mathcal{SP}^{\rightarrow} = tmp\mathcal{SP}^{\rightarrow}$;
$\mathcal{SP}^{\rightarrow}(end) = 0$;

Algorithm 4: Augmenting tree $\mathcal{A}$ with a tree $\mathcal{B}$

Input : $\mathcal{A} = \{\mathcal{A}^{\cdot}, \mathcal{A}^{\rightarrow}\}$, $\mathcal{A}_{index} = \{\mathcal{A}^{\cdot}_{index}, \mathcal{A}^{\rightarrow}_{index}\}$
$\mathcal{B} = \{\mathcal{B}^{\cdot}, \mathcal{B}^{\rightarrow}\}$, $\mathcal{B}_{index} = \{\mathcal{B}^{\cdot}_{index}, \mathcal{B}^{\rightarrow}_{index}\}$
Output: $\mathcal{A}$ with augmented vertices and indices of $\mathcal{B}$

for $i = \mathcal{B}^{\cdot}(first) \rightarrow \mathcal{B}^{\cdot}(last)$ **do**
  Set $newbranch = -1$;
  for $j = \mathcal{A}^{\rightarrow}_{index}(first) \rightarrow \mathcal{A}^{\rightarrow}_{index}(last)$ **do**
    Set $counter = 0$;
    for $k = \mathcal{A}^{\rightarrow}_{index}(j, first) \rightarrow \mathcal{A}^{\rightarrow}_{index}(j, last)$ **do**
      if $\mathcal{A}^{\rightarrow}(j,k) = \mathcal{B}^{\cdot}(i)$ **then** $newbrach = j$;
      else $counter++$;
    end
  end
  if $newbrach > -1$ **then**
    Set $aug^{\cdot} = \mathcal{B}^{\cdot}(i)$ and $aug^{\rightarrow} = \mathcal{A}^{\rightarrow}(newbrach)$;
    $\mathcal{A}^{\rightarrow}(newbrach) = \mathcal{B}^{\rightarrow}(i)$;
    Insert $aug^{\cdot}$ into $A^{\cdot}$ after element $newbrach$;
    Insert $aug^{\rightarrow}$ into $A^{\rightarrow}$ after element $newbrach$;
    Set $aug^{\cdot}_{index} = \mathcal{B}^{\cdot}_{index}(i)$ and $aug^{\rightarrow}_{index} = \mathcal{A}^{\rightarrow}_{index}(newbrach)$;
    Remove the elements of $aug^{\rightarrow}_{index}$ until element $counter$;
    Remove the elements of $\mathcal{A}^{\rightarrow}_{index}$ after element $counter$;
    Insert $aug^{\cdot}_{index}$ into $\mathcal{A}^{\cdot}_{index}$ after element $newbrach$;
    Insert $aug^{\rightarrow}_{index}$ into $\mathcal{A}^{\rightarrow}_{index}$ after element $newbrach$;
end

3.1.4.4 Merging augmented trees

Augmented trees are merged using the same idea that is suggested in [21]. The algorithm is given in Algorithm 5. In the beginning, two new arrays for the up- and down-degrees of the vertices are initialized. Only the vertices in the augmented join tree with 0 up-degree and 1 down-degree or the vertices in the augmented split tree with 0 down-degree and 1 up-degree are processed. When a vertex is found and processed, it is removed from the list and corresponding up- and down-degrees are updated.

In the last part of the algorithm, it is confirmed that all vertices are directed to some vertex in the graph. This makes sure that all the pixels are included in the indices lists. Basically, in this part, if a vertex is found to be not directed to any other, a new edge is appended at the end which directs it to itself.

Algorithm 5: Merging augmented trees

Input : Augmented join and split trees $\mathcal{JA}$, $\mathcal{JA}_{index}$, $\mathcal{SA}$, $\mathcal{SA}_{index}$
Output: Merged tree $\mathcal{M} = \{\mathcal{M}^{\cdot}, \mathcal{M}^{\rightarrow}\}$, $\mathcal{M}_{indices} = \{\mathcal{M}^{\cdot}_{indices}, \mathcal{M}^{\rightarrow}_{indices}\}$

Initialize $\mathcal{M}^{\cdot} = \{\}, \mathcal{M}^{\rightarrow} = \{\}, \mathcal{M}^{\cdot}_{indices} = \{\}, \mathcal{M}^{\rightarrow}_{indices} = \{\}$;
Initialize $\delta^{+} = \{\}$ and $\delta^{-} = \{\}$;
$\forall i$ set $\delta^{+}(i)$ as the number of $\mathcal{JA}^{\cdot}(i)$ in $\mathcal{JA}^{\rightarrow}$;
$\forall i$ set $\delta^{-}(i)$ as the number of $\mathcal{SA}^{\cdot}(i)$ in $\mathcal{SA}^{\rightarrow}$;
while $\exists \delta^{+} \neq 0$ *or* $\exists \delta^{-} \neq 0$ **do**
  for $i = \mathcal{JA}^{\cdot}(first) \rightarrow \mathcal{JA}^{\cdot}(last)$ **do**
    if $\delta^{+}(i) = 0$ & $\mathcal{JA}^{\cdot}(i) \neq 0$ **then**
      Find j: $\mathcal{SA}^{\cdot}(j) = \mathcal{JA}^{\cdot}(i)$;
      if $\delta^{-}(j) = 1$ **then**
        $\delta^{-}(j) = \delta^{-}(j) - 1$;
        $\mathcal{M}^{\cdot}(\mathcal{JA}^{\cdot}(i)) = \mathcal{JA}^{\cdot}(i)$; $\mathcal{M}^{\rightarrow}(\mathcal{JA}^{\cdot}(i)) = \mathcal{JA}^{\rightarrow}(i)$;
        $\mathcal{M}^{\cdot}_{index}(\mathcal{JA}^{\cdot}(i)) = \mathcal{JA}^{\cdot}_{index}(i)$; $\mathcal{M}^{\rightarrow}_{index}(\mathcal{JA}^{\cdot}(i)) = \mathcal{JA}^{\rightarrow}_{index}(i)$;
        Find l: $\mathcal{JA}^{\cdot}(l) = \mathcal{JA}^{\rightarrow}(i)$;
        $\delta^{+}(l) = \delta^{+}(l) - 1$; $\mathcal{JA}^{\cdot}(i) = \mathcal{JA}^{\rightarrow}(i) = 0$;
        Trace in $\mathcal{SA}$ the vertices $\mathcal{SA}^{\cdot}(j) = \mathcal{JA}^{\rightarrow}(i)$ until $\mathcal{JA}^{\cdot}(i)$;
        Form a list of the traced vertices, tr;
        $\mathcal{SA}^{\rightarrow}(tr(end)) = \mathcal{SA}^{\rightarrow}(j)$; $\mathcal{SA}^{\cdot}(j) = \mathcal{SA}^{\rightarrow}(j) = 0$;
        Update all $\mathcal{SA}^{\rightarrow}_{index}(tr) = \mathcal{SA}^{\rightarrow}_{index}(tr) \setminus \mathcal{JA}^{\rightarrow}_{index}(i)$
  end
  for $i = \mathcal{SA}^{\cdot}(first) \rightarrow \mathcal{SA}^{\cdot}(last)$ **do**
    if $\delta^{-}(i) = 0$ & $\mathcal{SA}^{\cdot}(i) \neq 0$ **then**
      Find j: $\mathcal{JA}^{\cdot}(j) = \mathcal{SA}^{\cdot}(i)$;
      if $\delta^{+}(j) = 1$ **then**
        $\delta^{+}(j) = \delta^{+}(j) - 1$;
        $\mathcal{M}^{\cdot}(\mathcal{SA}^{\cdot}(i)) = \mathcal{SA}^{\cdot}(i)$; $\mathcal{M}^{\rightarrow}(\mathcal{SA}^{\cdot}(i)) = \mathcal{SA}^{\rightarrow}(i)$;
        $\mathcal{M}^{\cdot}_{index}(\mathcal{SA}^{\cdot}(i)) = \mathcal{SA}^{\cdot}_{index}(i)$; $\mathcal{M}^{\rightarrow}_{index}(\mathcal{SA}^{\cdot}(i)) = \mathcal{SA}^{\rightarrow}_{index}(i)$;
        Find l: $\mathcal{SA}^{\cdot}(l) = \mathcal{SA}^{\rightarrow}(i)$;
        $\delta^{+}(l) = \delta^{+}(l) - 1$; $\mathcal{SA}^{\cdot}(i) = \mathcal{SA}^{\rightarrow}(i) = 0$;
        Trace in $\mathcal{JA}$ the vertices $\mathcal{JA}^{\cdot}(j) = \mathcal{SA}^{\rightarrow}(i)$ until $\mathcal{SA}^{\cdot}(i)$;
        Form a list of the traced vertices, tr;
        $\mathcal{JA}^{\rightarrow}(tr(end)) = \mathcal{JA}^{\rightarrow}(j)$; $\mathcal{JA}^{\cdot}(j) = \mathcal{JA}^{\rightarrow}(j) = 0$;
        Update all $\mathcal{JA}^{\rightarrow}_{index}(tr) = \mathcal{JA}^{\rightarrow}_{index}(tr) \setminus \mathcal{SA}^{\rightarrow}_{index}(i)$
  end
end

```
for i = 1 → max(JA^·) do
    if i ∉ M^· then
        if i ∈ M^→ then
            Find j : JA^·(j) = i;
            Append i to M^· and M^→;
            Append JA^·_index(j) to M^·_index and {} to M^→_index;
        else
            Find j : SA^·(j) = i;
            Append i to M^· and M^→;
            Append SA^·_index(j) to M^·_index and {} to M^→_index;
end
```

3.1.4.5 Pruning

For Morse functions the output of the merging given in Algorithm 5 is the contour tree. However, non-Morse functions require a compulsory pruning operation that is followed by the merging of the augmented trees. Pruning combines the degenerate critical points that are connected and constructs contour trees for non-Morse functions. Also for both Morse and non-Morse functions, pruning cleans up insignificant vertices and edges.

More commonly, contour trees for non-Morse functions are calculated after perturbation, which removes degeneracies other than multisaddles. Perturbation is generally done using the Simulation of Simplicity method [41], where a very small number is added to each data point according to its location in physical memory. Perturbation renders out the need for combining degenerate critical points that are connected in the pruning step and decreases the computational cost. However, a contour tree given in Definition 17 cannot be obtained once the data is perturbed.

There have been few approaches suggested in the literature to perform pruning for contour trees. In [4, 24], insignificant vertices and edges are determined and removed by using geometrical measures such as surface area or volume from contour trees. In [141, 142], branches are assigned importance values which are calculated by a function of three parameters, p, v and hv. p stands for persistence, which is defined as the absolute difference between intensities at the vertices of an edge, v is the volume and hv is the hypervolume, which is defined as the integral of the scalar field on an edge. Based on the importance value of an edge it is either kept or removed.

In order to further simplify the visualization of contour trees and represent them in more intuitive ways, several approaches are suggested such as the topological landscapes [38, 48, 81, 131], the toporrery [87] and topological cacti [132].

In this thesis, three graph operations (i) *vertex collapse*, (ii) *leaf pruning* and (iii) *saddle collapse* are used to prune merge trees. These operations and in which cases they are applied are shown in Fig.3.8.

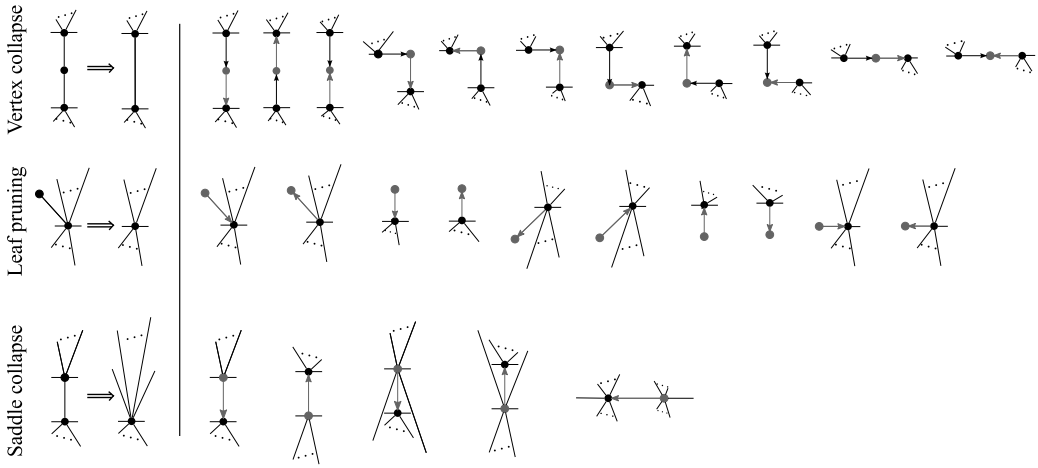


Figure 3.8: Graph operations used for pruning. On the first column, the main operations of vertex collapse, leaf pruning and saddle collapse are shown. The second column shows all the cases where the operations are applied. The insignificant vertices and edges are shown in gray.

Vertex collapse operation is done whenever possible. In order to prune leaves or collapse saddles, similar to [24] and [141], two inputs called the (i) *height threshold* and (ii) *area/volume threshold* are used to determine whether vertices are insignificant and will be removed. If $\mathcal{M}^{\cdot}(x)$ is a vertex of the merged tree, it is insignificant, either if $\mathcal{M}^{\cdot}_{index}(x) \cup \mathcal{M}^{\rightarrow}_{index}(x)$ has fewer elements than the area/volume threshold or the intensity difference $|I(\mathcal{M}^{\cdot}_{index}(x)) - I(\mathcal{M}^{\rightarrow}_{index}(x))|$ is smaller than the height threshold. The pseudo algorithm of the pruning is given in Algorithm 6.

Algorithm 6: Pruning merged trees

Input : Input image I, height threshold hT, area/volume threshold avT, $\mathcal{M} = \{\mathcal{M}^{\cdot}, \mathcal{M}^{\rightarrow}\}$, $\mathcal{M}_{index} = \{\mathcal{M}^{\cdot}_{index}, \mathcal{M}^{\rightarrow}_{index}\}$
Output: Contour tree $\mathcal{CT} = \{\mathcal{CT}^{\cdot}, \mathcal{CT}^{\rightarrow}\}$, $\mathcal{CT}_{indices} = \{\mathcal{CT}^{\cdot}_{indices}, \mathcal{CT}^{\rightarrow}_{indices}\}$

repeat
  for $i = \mathcal{M}^{\cdot}(first) \rightarrow \mathcal{M}^{\cdot}(last)$ **do**
    if *vertex collapse* **then**
      Find $j\colon \mathcal{M}^{\rightarrow}(j) = \mathcal{M}^{\cdot}(i)$ and $k\colon \mathcal{M}^{\cdot}(k) = \mathcal{M}^{\rightarrow}(i)$;
      $\mathcal{M}^{\rightarrow}(j) = \mathcal{M}^{\cdot}(k)$ and $\mathcal{M}^{\rightarrow}_{index}(j) = \mathcal{M}^{\rightarrow}_{index}(j) \cup \mathcal{M}^{\cdot}_{index}(i) \cup \mathcal{M}^{\rightarrow}_{index}(i)$;
      Remove $\mathcal{M}^{\cdot}(i), \mathcal{M}^{\rightarrow}(i), \mathcal{M}^{\cdot}_{index}(i), \mathcal{M}^{\rightarrow}_{index}(i)$;
    else if *leaf pruning* **then**
      Find $k\colon \mathcal{M}^{\cdot}(k) = \mathcal{M}^{\rightarrow}(i)$;
      if $|I(\mathcal{M}^{\cdot}_{index}(i)) - I(\mathcal{M}^{\cdot}_{index}(k))| \leq hT$ *or* $\#(\mathcal{M}^{\cdot}_{index}(i) + \mathcal{M}^{\rightarrow}_{index}(k)) \leq avT$ **then**
        $\mathcal{M}^{\cdot}_{index}(k) = \mathcal{M}^{\cdot}_{index}(k) \cup \mathcal{M}^{\cdot}_{index}(i) \cup \mathcal{M}^{\rightarrow}_{index}(i)$;
        Remove $\mathcal{M}^{\cdot}(i), \mathcal{M}^{\rightarrow}(i), \mathcal{M}^{\cdot}_{index}(i), \mathcal{M}^{\rightarrow}_{index}(i)$;
    else if *saddle collapse* **then**
      Find all $j\colon \mathcal{M}^{\rightarrow}(j) = \mathcal{M}^{\cdot}(i)$ and $k\colon \mathcal{M}^{\cdot}(k) = \mathcal{M}^{\rightarrow}(i)$;
      if $|I(\mathcal{M}^{\cdot}_{index}(i)) - I(\mathcal{M}^{\cdot}_{index}(k))| \leq hT$ *or* $\#(\mathcal{M}^{\cdot}_{index}(i) + \mathcal{M}^{\rightarrow}_{index}(k)) \leq avT$ **then**
        For all j, set $\mathcal{M}^{\rightarrow}(j) = \mathcal{M}^{\cdot}(k)$;
        $\mathcal{M}^{\cdot}_{index}(k) = \mathcal{M}^{\cdot}_{index}(k) \cup \mathcal{M}^{\cdot}_{index}(i) \cup \mathcal{M}^{\rightarrow}_{index}(i)$;
        Remove $\mathcal{M}^{\cdot}(i), \mathcal{M}^{\rightarrow}(i), \mathcal{M}^{\cdot}_{index}(i), \mathcal{M}^{\rightarrow}_{index}(i)$;
  end
until *no change*;
$\mathcal{CT} = \mathcal{M}$;
$\mathcal{CT}_{index} = \mathcal{M}_{index}$;

3.2 Representation of binary images using contour trees

It is described in subsection 3.1.1 that contour trees are generally used to simplify and help visualize scalar fields. In this section, a different application of contour trees where it is used to represent binary images is explained. This is done simply by using the Euclidean distance transform (EDT) which transforms a binary image to a scalar image, this approach is first presented in [Publication-I].

EDT is a fundamental geometrical operator which has been used often in computer vision, pattern recognition, computational geometry and shape analysis applications [84]. Distance transform is first mentioned in [94] as a mapping of each element into its smallest distance to a given subset. Here the minimum distance of all elements to the boundary

of the foreground is considered. The formal expression of the signed EDT used in this thesis is given as follows:

Definition 18 (Euclidean distance transform (EDT))**.** *EDT is the following mapping from a binary image $I(X)$ to an image $EDT(X)$.*

$$EDT(\mathbf{x}) = \begin{cases} min\{\| \mathbf{x}-\mathbf{y} \|\} \ for\ \mathbf{x} \in F\ \&\ \forall \mathbf{y} \in F^b \\ 0 \ for\ \mathbf{x} \in F^b \\ -min\{\| \mathbf{x}-\mathbf{y} \|\} \ for\ \mathbf{x} \in F^c \& \ \forall \mathbf{y} \in F^b \end{cases} \tag{3.4}$$

$\| \mathbf{x}-\mathbf{y} \| = \sqrt{\sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{y}_k)^2}$, *the Euclidean distance between* $\mathbf{x}$ *and* $\mathbf{y}$.

There are several algorithms proposed to calculate EDT. A review of the most common approaches can be found in [42]. In this study, the popular algorithm of Maurer's is used, which is based on dimensionality reduction and partial Voronoi diagram construction [69].

Fig.3.9 shows an example of a contour tree representation for a binary image. Representation of binary images using contour trees reveal important geometrical and topological information. Above the zero level, information about the number of connected components and their geometry is given. Below the zero level, the geometry of background is represented which provides information about how connected components are placed in the image. The foreground and background representations reveal how thick connected components get and the separation between them.

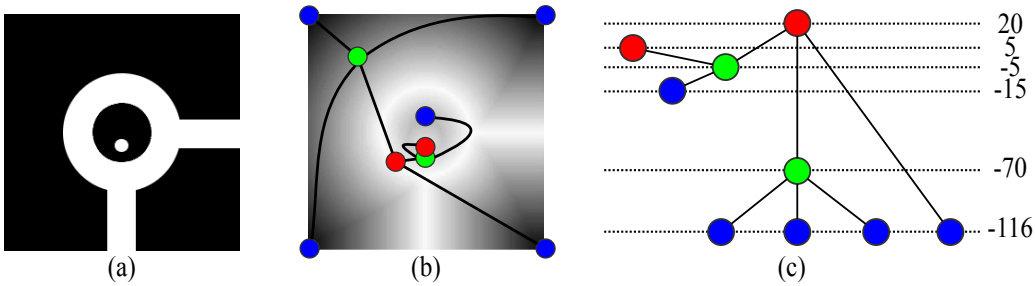
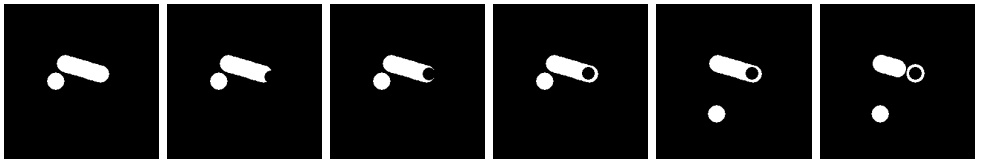


Figure 3.9: Binary image representation using contour trees. (a) Binary input image (b) contour tree drawn on top of the EDT of the input (c) contour tree and levels of vertices

3.3 Contour tree connectivity

In this section, the main contribution of the thesis, the contour tree connectivity (CTC) is introduced. CTC is featured in all the publications except [Publication-I].

CTC offers an alternative connectivity measure for binary images by providing connectivity information that takes into account the proximity of connected components and cavities. For CTC to be easy to use, it is designed to be a real number between 0 and 1. For example, Fig.3.10 shows that there is an 11% connectivity decrease in terms of CTC from left to right. Decreases occur due to cavity formation (a-c), hole formation (d), separation of connected components (e) and increase in the number of connected components (f). In contrast, Euler number only varies when a topology change occurs in (d) and (f).



CTC=0.71 χ=2	CTC=0.70 χ=2	CTC=0.68 χ=2	CTC=0.64 χ=1	CTC=0.62 χ=1	CTC=0.60 χ=2
(a)	(b)	(c)	(d)	(e)	(f)

Figure 3.10: The connectivity information provided by CTC shows a total of 11% connectivity decrease from left to right. CTC is decreasing due to cavity formation (a-c), hole formation (d), separation of connected components (e) and increase in the number of connected components (f). Notice that Euler number varies only in (d) and (f) where the topology changes. (Reproduced from [Publication-II].)

The main idea behind CTC is to compute the structural connectivity of a binary image by computing the connectivity of a variant of its contour tree representation. Fig.3.11 shows the flowchart for extracting CTC. Starting with a binary image, firstly a real valued image is obtained using EDT. The contour tree of this image is computed and later processed to obtain a new tree representation which is called the supplemented contour tree (SCT). Lastly, connectivity of SCT is calculated using algebraic connectivity and normalized with a function of the range of the EDT image, which gives the CTC.

Before describing what SCT is and why there is a need for SCT, in the next subsection, information about algebraic graph connectivity is given.

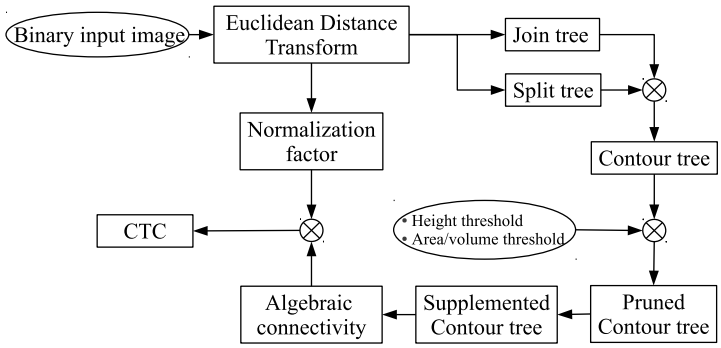


Figure 3.11: Flowchart for the CTC extraction. From a binary input image, firstly EDT, then the SCT and finally the CTC is calculated as the normalized algebraic connectivity of SCT. (Reproduced from [Publication-II].)

3.3.1 Graph spectrum and algebraic graph connectivity

Spectrum of graphs has been useful for many applications in computer science [31]. For instance, the Googlebot search engine uses ideas derived from spectral graph theory [17]. Graph spectrum and algebraic connectivity of graphs have also found several applications in image processing [34, 107, 135].

For a graph $\mathcal{G}$, the eigenvalues of the Laplacian matrix given as $L(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G})$, form the spectrum of $\mathcal{G}$. Here $D(\mathcal{G})$ and $A(\mathcal{G})$ are the degree diagonal and the adjacency matrices respectively. Note that for a connected graph, the eigenvalues of $L(\mathcal{G})$ can be sorted as $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. Since $\lambda_1 = 0$ for disconnected graphs, Fiedler gave the following definition for algebraic connectivity in [43] as a measure of graph connectivity:

Definition 19. *Algebraic connectivity of a graph $\mathcal{G}$, $\lambda_1(\mathcal{G})$, is the second smallest eigenvalue of its Laplacian matrix.*

3.3.2 Supplemented contour tree

In section 3.2, it is shown that binary image representation using contour trees reveal important geometrical and topological properties such as information about the connected components, their thickness and separations. However, this information is possible, provided that the indices for the vertices are used together with the input image. As for the contour tree itself, it is only a graph with a set of vertices and edges which does not contain the intensity levels which vertices represent. Therefore as only graphs, contour trees

give very high level of abstractions whose connectivities do not provide discriminative information. Additionally, in order to compare connectivities of different binary images, there is a need to normalize algebraic connectivity values since sorting trees with respect to their algebraic values is challenging. Supplemented contour tree (SCT) is designed to create a graph representation from a contour tree with lesser abstraction and more discrimination. Additionally, SCT makes it easier to normalize algebraic connectivity which is explained in the next subsection.

SCT is obtained by introducing additional vertices into a contour tree. Vertices are added to represent the intensity differences between the ends of edges by the number of new vertices in between them. This addition of new vertices is named *supplementation* thus the name supplemented contour tree. SCT is obtained using Algorithm 7.

Algorithm 7: Computing the supplemented contour tree

Input : Input image I, $\mathcal{CT} = \{\mathcal{CT}^{\cdot}, \mathcal{CT}^{\rightarrow}\}$, $\mathcal{CT}_{index} = \{\mathcal{CT}^{\cdot}_{index}, \mathcal{CT}^{\rightarrow}_{index}\}$
Output: Supplemented contour tree $\mathcal{SCT} = \{\mathcal{SCT}^{\cdot}, \mathcal{SCT}^{\rightarrow}\}$

Initialize $\mathcal{SCT}^{\cdot} = \mathcal{SCT}^{\rightarrow} = \{\}$;
Set $newV = \max(\mathcal{CT}^{\cdot}) + 1$;
for $i = \mathcal{CT}^{\cdot}(first) \rightarrow \mathcal{CT}^{\cdot}(last)$ **do**
    Find k: $\mathcal{CT}^{\rightarrow}(i) = \mathcal{CT}^{\cdot}(k)$;
    $startLevel = I(\mathcal{CT}^{\cdot}_{index}(i))$;
    $endLevel = I(\mathcal{CT}^{\cdot}_{index}(k))$;
    $dif = |startLevel - endLevel|$;
    if $dif > 1$ **then**
        Append $\mathcal{CT}^{\cdot}(i)$ to $\mathcal{SCT}^{\cdot}$;
        Append $newV$ to $\mathcal{SCT}^{\rightarrow}$;
        for $j = (startLevel - sign(dif)) \rightarrow (endLevel + 2sign(dif))$ **do**
            $newV = newV + 1$;
            Append $newV - 1$ to $\mathcal{SCT}^{\cdot}$; `// Supplement vertex`
            Append $newV$ to $\mathcal{SCT}^{\rightarrow}$; `// Supplement edge`
        end
        Append $newV$ to $\mathcal{SCT}^{\cdot}$;
        Append $\mathcal{CT}^{\rightarrow}(i)$ to $\mathcal{SCT}^{\rightarrow}$;
        $newV = newV + 1$;
    else
        Append $\mathcal{CT}^{\cdot}(i)$ to $\mathcal{SCT}^{\cdot}$;
        Append $\mathcal{CT}^{\rightarrow}(i)$ to $\mathcal{SCT}^{\rightarrow}$;
end

SCT is a useful graph representation for binary images as it is capable of showing how foreground/background regions are positioned in the image by the number of vertices along edges. The example in Fig. 3.12 shows the difference between the contour tree and the SCT.

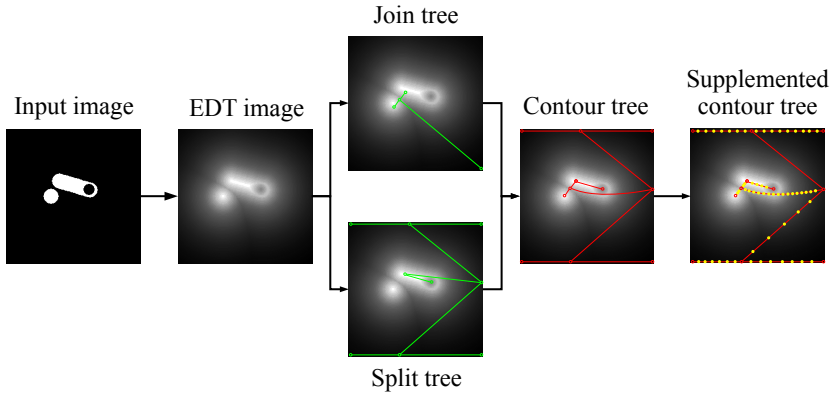


Figure 3.12: Binary image representation using the supplemented contour trees. First EDT, then join and split trees (green trees) are calculated. Join and split trees are merged to form the contour tree (red tree). SCT is shown on the right (red tree with yellow supplemented nodes). (Reproduced from [Publication-II].)

The vertices of a contour tree where topological changes occur are called *supernodes* [22]. Notice that the contour tree definition given in Definition 17 only contains supernodes. Contour trees which only contain supernodes are called *unaugmented* contour trees in the literature [22]. An *augmented* contour tree can be obtained by introducing vertices from other significant levels into the graph. However in contrast to an augmented contour tree, the vertices that are introduced in SCT do not necessarily represent contours of the input image.

3.3.3 Normalization - Upper and lower bounds

One problem regarding algebraic connectivity is associated with the bounds. For example, [11] gives a lower bound of $\lambda_1 \geq 4/(n.diam)$. This implies that in the set of SCTs with the same diameter, graphs having larger number of vertices, n, have lower λ_1 values. However, this might not necessarily be a good comparative measure for connectivity. The problem with λ_1 bounds makes ordering graphs using algebraic connectivity a challenging

research problem in spectral graph theory. A review of the recent studies concerning algebraic connectivity, bound values and ordering graphs based on λ_1 is given in [65].

In order to compare algebraic connectivities of SCTs belonging to different binary images, the height of SCT is used as the parameter for normalization. Using tree heights not only solves the normalization problem with a simple function but also makes it possible to fairly compare connectivities of SCTs with different number of vertices. The normalization based on SCT heights is done using the following lower and upper bounds.

From subsection 3.3.1 and Definition 19, we give the following lower bound:

Corollary 3. *$\mathcal{G}$ being an SCT, $\lambda_1(\mathcal{G}) > 0$ is a lower bound.*

The upper bound for λ_1 is given by the following theorem:

Theorem 3. *$\mathcal{G}$ being an SCT, $\lambda_1(\mathcal{G}) \leq 2(1 - \cos\frac{\pi}{h+1})$ is an upper bound, where h is the height of $\mathcal{G}$.*

Proof. We use the following corollary derived from Theorem 6.4.1 (page 240) and Corollary 6.4.3 (page 242) given in [73].

Corollary 4. *Suppose that $\hat{\mathcal{G}}$ is a subtree of the unweighted tree $\mathcal{G}$. Then $\lambda(\hat{\mathcal{G}}) > \lambda(\mathcal{G})$.*

Let $\mathcal{G}$ be any SCT. Pick $\hat{\mathcal{G}}$ as any path from any vertex on $\lfloor l^g_{max} \rceil$ to any vertex on $\lfloor l^g_{min} \rceil$. Note that $\mathcal{G}$ and $\hat{\mathcal{G}}$ have the same root vertex and height, h. Then $L(\hat{\mathcal{G}})$ is the following $(h+1) \times (h+1)$ matrix:

$$L(\hat{\mathcal{G}}) = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}_{(h+1)\times(h+1)} \tag{3.5}$$

which has the eigenvalues $\lambda_k = 2(1 - \cos\frac{k\pi}{h+1})$, $k = \{0, 1, \cdots, h\}$ [136]. Therefore, from Corollary 4, $\lambda_1(\mathcal{G}) \leq \lambda_1(\hat{\mathcal{G}}) = 2(1 - \cos\frac{\pi}{h+1})$. □

3.3.4 Formal definition of contour tree connectivity

Finally, by using Definition 19, Corollary 3 and Theorem 3, the formal definition of the CTC is given as follows:

Definition 20. *(Contour tree connectivity (CTC)) CTC of a binary image I is:*

$$\overline{\lambda_1(\mathcal{G})} = \frac{\lambda_1(\mathcal{G})}{2\left[1 - \cos\left(\frac{\pi}{\lfloor l^g_{max} \rceil - \lfloor l^g_{min} \rceil + 1}\right)\right]} \tag{3.6}$$

where $\mathcal{G}$ is the SCT of I, λ_1 is the algebraic connectivity of $\mathcal{G}$, l^g_{max} and l^g_{min} are the global max and min of EDT of I.

In the rest of the text, the abbreviation CTC and the symbol $\overline{\lambda_1}$ are used interchangeably. Lastly, we give the following corollary and conclude this chapter.

Corollary 5. *For any binary image, $0 < \overline{\lambda_1} \leq 1$.*

4 EXPERIMENTS AND RESULTS

This chapter is divided into 3 sections: Case A, B and C. In Case A, we experiment with synthetic 2D and 3D toy images. The main goal is to introduce the connectivity interpretation of CTC. Case B and C considers applications of CTC. In Case B, we analyze trabecular bone connectivity and how it relates to its mechanical strength. In Case C, we study flow in simple microstructures and how this is related with connectivity.

4.1 Case A: Tests on 2D and 3D binary images

In this section, we study contour tree connectivity (CTC) using 2D and 3D binary images [Publication-II]. This section aims to show how the theory behind CTC works and what kind of connectivity information is provided by it. In order to demonstrate the intuition behind CTC, in subsection 4.1.1, 2D images are studied where carefully designed series of images are analyzed. In subsection 4.1.2, 3D images are used to show a simple example to how CTC can be used to study complex structures. In the last subsection of 4.1.3, the results of the computation speed analysis for our implementation is given.

4.1.1 2D test images

We tested CTC with the synthetic dataset shown in Fig.4.1. The dataset is composed of 6 series (I-VI). Each series contains 6 images (A-F), that have slight changes in connectivity. All images are of dimensions 256×256 pixels. We prepared different cases for each series to show how $\overline{\lambda_1}$ varies. For example, series II shows the changes in $\overline{\lambda_1}$ during clumping of objects, III shows the changes during branching, IV shows the effect of holes.

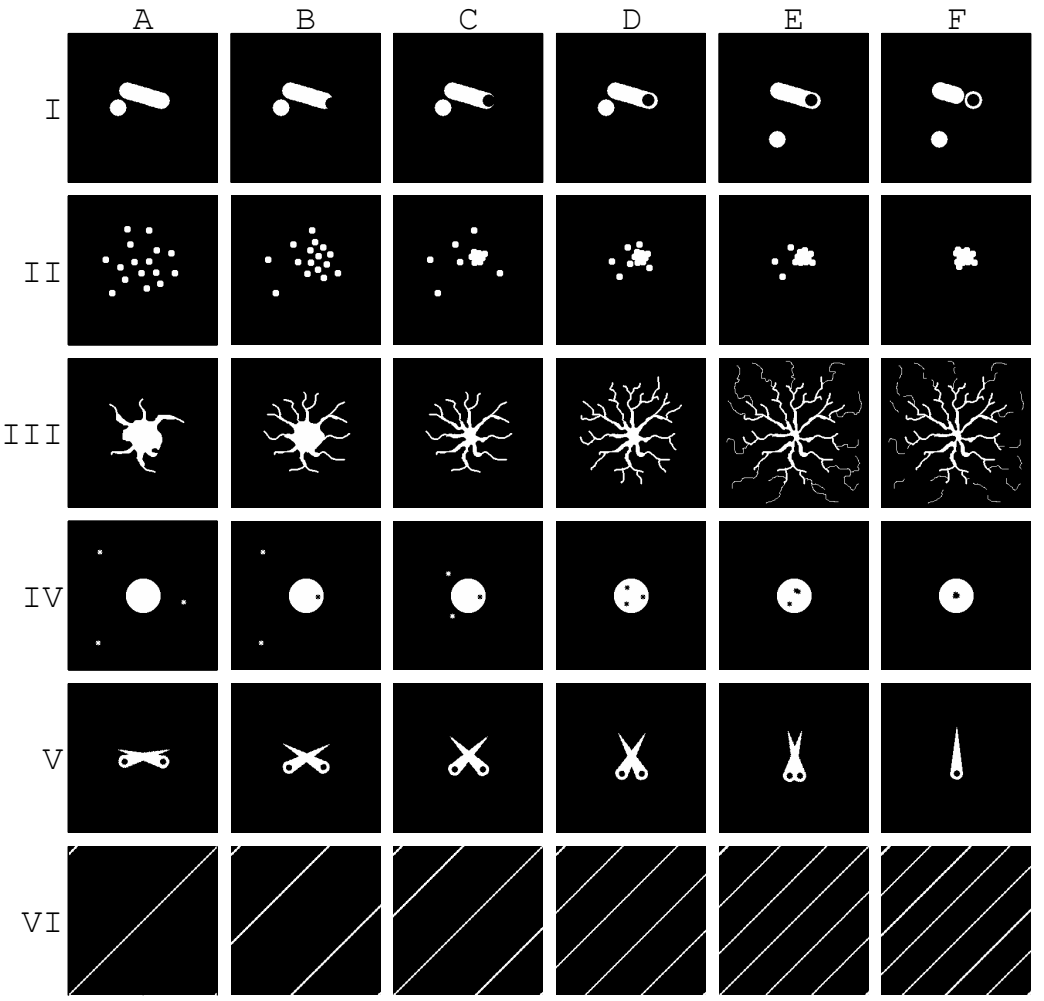


Figure 4.1: 2D images used in the experiments. Each row contains a series of images. In each column connectivity is slightly altered. (Reproduced from [Publication-II].)

We also tested CTC using the same images under various boundary noise conditions. To add boundary noise, we randomly moved the boundary pixels. We considered Gaussian noise with $\mu = 0$ and $\sigma = \{1, 2, 3, 4\}$. Samples from the noisy images are shown in Fig.4.2. This type of noise allows for topological changes that significantly alter Euler number which is shown in Fig.4.3.

Fig.4.4 shows the $\overline{\lambda_1}$ values for the images. Error bars show the max and min values of $\overline{\lambda_1}$ for noisy images. It is observed from Fig.4.4 that $\overline{\lambda_1}$ consistently increases when foreground objects approach each other. This trend is best seen for series II. Also the

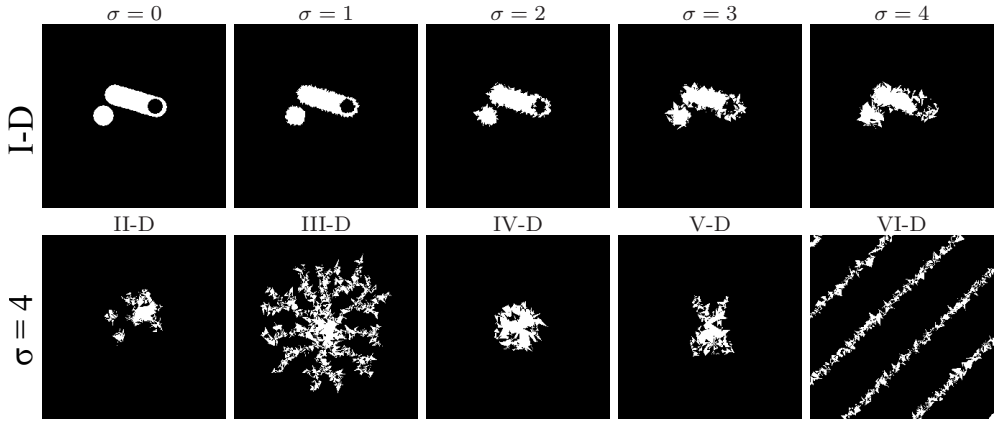


Figure 4.2: Top row shows various levels of noise on image I-D. Bottom row shows the maximum noise for images II-D to VI-D. (Reproduced from [Publication-II].)

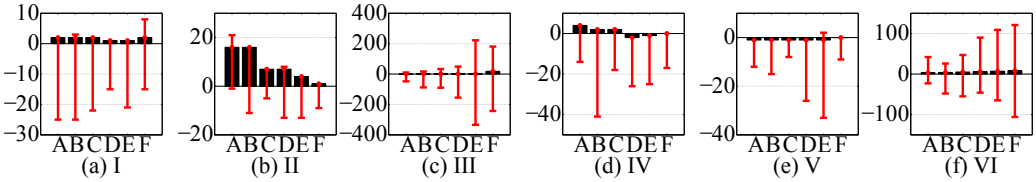
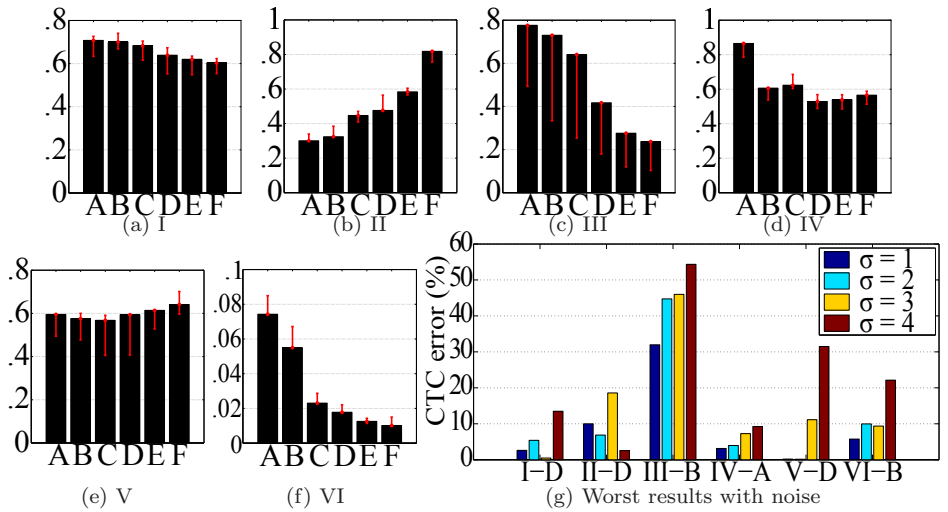


Figure 4.3: Bar graphs in (a)-(f) show the calculated Euler numbers for the 2D dataset. The error bars show the maximum and minimum values of Euler number in the presence of noise.

changes from I-D to I-E and IV-B to IV-C occur due to the same reason. The decreasing trend in $\overline{\lambda_1}$ during branching is seen best in series III. From I-A to I-B and I-C, a decreasing trend is present during cavity formation. When cavities turn into holes, the decreasing $\overline{\lambda_1}$ trend continues, for example in I-D. It is also observed from series IV that the decrease in the number of holes increases $\overline{\lambda_1}$. From series VI, it is seen that increasing details decreases $\overline{\lambda_1}$. Lastly, disconnecting objects decrease $\overline{\lambda_1}$ as seen in I-F and II-F.

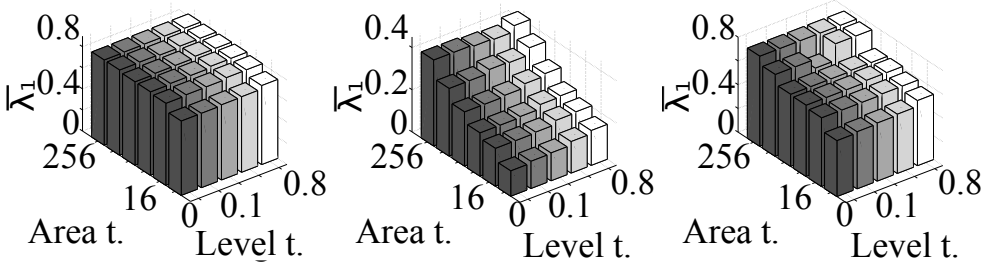
We picked the worst results to show the effect of noise. For each series, the image with worst error is picked and the absolute error is plotted in Fig.4.4g for all noise conditions. From Fig.4.4g, it is observed that for $\sigma = \{1, 2\}$, $\overline{\lambda_1}$ change is mostly below 10%. For I-D, IV-A and VI-B, increasing the noise to $\sigma = 3$ still keeps $\overline{\lambda_1}$ within 10% error. In the case of extreme noise ($\sigma = 4$), for most cases $\overline{\lambda_1}$ changes drastically. CTC performs bad for series III where very thin branches exist. This is partly due to how we added the noise. Fig.4.2 shows the effect of this type of noise on branches of III-D.



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) Worst results with noise

Figure 4.4: Bar graphs in (a)-(f) show the calculated $\overline{\lambda_1}$ values for the 2D dataset. The error bars show the maximum and minimum values of $\overline{\lambda_1}$ in the presence of noise. Worst cases of error are shown in (g) for varying noise. (Reproduced from [Publication-II].)

We picked one good, one bad and one typical case to show the effect of the pruning parameters on CTC. Fig.4.5 shows the changes for various combinations of level and area thresholds on images IV-A, III-D and V-A. It is observed that pruning does not alter $\overline{\lambda_1}$ significantly for image IV-A despite the noise. Whereas for III-D, pruning parameters affect the $\overline{\lambda_1}$ value. On a typical case, pruning parameters do alter $\overline{\lambda_1}$. However, it can be observed from Fig.4.5 that there is a flat region in a typical case which makes it easy to decide on the parameters. For example, for all results in this subsection, a level threshold of 0.1 and an area threshold of 16 is used regardless of the data or the noise.



(a) Good case: IV-A with noise $\sigma = 4$ (b) Bad case: III-D with noise $\sigma = 4$ (b) Typical case: V-A with noise $\sigma = 1$

Figure 4.5: Bar graphs showing the effects of pruning parameters on (a) good (b) bad and (c) typical cases. As default values, area threshold of 16 and level threshold of 0.1 is found to work reasonable. (Reproduced from [Publication-II].)

4.1.2 3D test images

Euler number is the standard measure to study connectivity of microstructures of materials [82]. In order to show the complementary value of CTC to identify microstructures, we used 3D artificial samples created in MATLAB® that resemble segmented microstructures obtained with a μCT device.

We prepared six groups of data. Each group contains 50 samples of dimensions 128×128×128 voxels. Three of the groups are made of spheres and the rest of the three are made of lines. We will abbreviate the sphere samples as S1, S2, S3 and line samples as L1, L2, L3.

All images are created using two parameters: the number of objects and the dimensions of objects. The object dimension for S1, S2 and S3 is the radius of spheres. For L1, L2 and L3, the object dimension is the width of lines. We used a Gaussian distributed random variable with mean $\mu = 32$ and standard deviation $\sigma = 6$ voxels for the length of lines in all L1, L2 and L3. For all samples, partial overlapping of objects was allowed. Fig.4.6 shows sample images from each group used in the test.

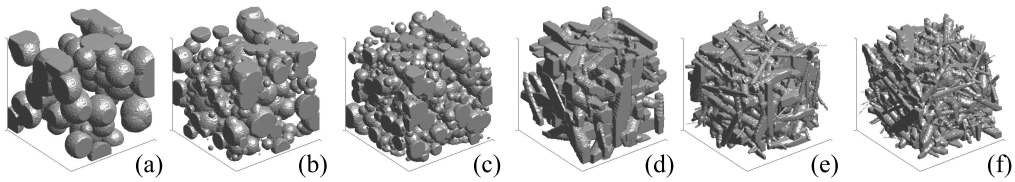


Figure 4.6: Samples from the 3D dataset used in the test. (a) S1, (b) S2, (c) S3, (d) L1, (e) L2 and (f) L3. (Reproduced from [Publication-II].)

For each group, we used a constant number of objects. For the dimensions of the objects, Gaussian distributed random variables are used. For S2 and L2, we used two random variables to create intermediate samples of S1-S3 and L1-L3. The parameters for all samples are set so that the volume ratio of foreground to background is ≈ 1. Table.4.1 shows the parameters that we used to generate the dataset.

Table 4.1: Parameters used to generate the dataset for the 3D test samples. The values in (. ,.) are the μ and σ of Gaussian distributed random variables used for setting object dimensions. The number of objects and the random variables used for S2 and L2 are shown in [. ,.]. (Reproduced from [Publication-II].)

	S1	S2	S3	L1	L2	L3
Number of objects	90	[45,400]	800	200	[100,400]	800
Dimension of objects	(16,4)	[(16,4),(6,4)]	(6,4)	(16,4)	[(16,4),(6,4)]	(6,4)

By using three features we performed a classification test. The used features are CTC ($\overline{\lambda_1}$), nominal opening dimension (d_{nom}) and Euler number (χ):

a) $\overline{\lambda_1}$*:* is calculated as explained in subsection 3.3.

b) d_{nom}*:* is calculated as explained in subsection 2.4.3.

c) χ*:* is calculated as explained in subsection 2.4.1.

To classify the samples, we used the naive Bayesian classifier because it is simple and known to achieve good results even in challenging multivariate cases [46]. Naive Bayesian classifier utilizes the Bayes' theorem to assign a probability that the input data is from a given class [40]. Let $C = \{c_1, \cdots, c_N\}$ be a set of classes c_i where N is the number of classes, $\{i \in \mathbb{N}^+ | i \leq N\}$ and let $\mathbf{x}$ be a $\mathcal{D}$ dimensional feature vector with independent features x_j, $\{j \in \mathbb{N}^+ | j \leq \mathcal{D}\}$. Then given the prior probability of observing the class $P(c_i)$, the probability of a particular class is:

$$P(c_i|\mathbf{x}) \propto P(c_i)P(\mathbf{x}|c_i) \propto P(c_i) \prod_{j=1}^{\mathcal{D}} P(x_j|c_i) \tag{4.1}$$

where $P(x_j|c_i)$ is the conditional probability of x_j given class c_i. Following Eq.4.1, Bayes' decision rule can be given as:

$$c = \arg\max_{c_i \in C} P(c_i|\mathbf{x}) \propto \arg\max_{c_i \in C} P(c_i) \prod_{j=1}^{\mathcal{D}} P(x_j|c_i) \tag{4.2}$$

In our classification problem, $N = 6$ and $\forall c_i \in C \; P(c_i) = 1/6$. Therefore, the feature vector $\mathbf{x}$ is assigned the class c_i; so that c_i maximizes $P(c_i|\mathbf{x})$ which is proportional to $\prod_{j=1}^{\mathcal{D}} P(x_j|c_i)$. Thus, $P(x_j|c_i)$ are required. This estimation is done using the MATLAB's built-in `NaiveBayes` class. We used half of the samples in each group for training and the other half for testing. Lastly, the classification accuracy, A, is calculated as follows:

$$A = \left(1 - \frac{\#\text{ of misclassified samples}}{\#\text{ of all samples}}\right) \times 100 \tag{4.3}$$

Classifications are done using all possible two combinations of the three features. In order to show what additional information comes with $\overline{\lambda_1}$, in Fig.4.7 we first give the estimated class conditional probabilities for each feature. Maximum peaks are normalized to 1 for easier visual comparison.

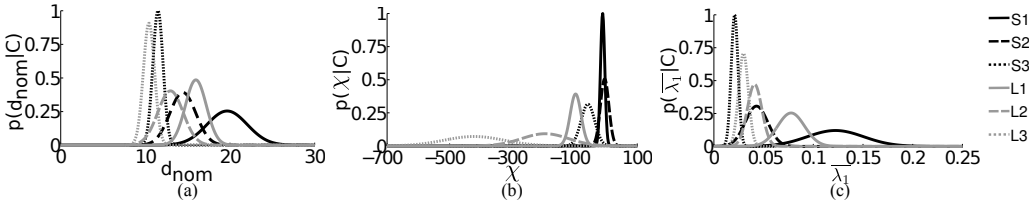


Figure 4.7: Class conditional probabilities. The legend shows the corresponding classes for different lines. From left to right, class conditional probabilities for (a) d_{nom}, (b) χ and (c) $\overline{\lambda_1}$ are plotted. (Reproduced from [Publication-II].)

The scatter plots are given for the feature combinations of $\{\chi, d_{nom}\}$, $\{\overline{\lambda_1}, d_{nom}\}$ and $\{\overline{\lambda_1}, \chi\}$ in Fig.4.8.

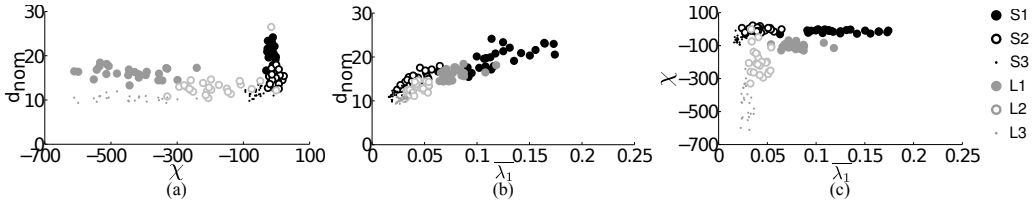


Figure 4.8: Scatter plots. The legend shows the corresponding classes for different markers. From left to right, scatter plots for feature sets (a) $\{\chi, d_{nom}\}$, (b) $\{\overline{\lambda_1}, d_{nom}\}$ and (c) $\{\overline{\lambda_1}, \chi\}$ are shown.

Table 4.2 shows the confusion matrices and the classification accuracies. Based on the distributions, a naive Bayesian classifier yields a 72.3% classification accuracy using only χ. With $\overline{\lambda_1}$ included, the accuracy increases to 92.7%.

Table 4.2: Confusion matrices. First column shows the true classes, the third row shows the predicted classes with $\{\chi, d_{nom}\}$, $\{\overline{\lambda_1}, d_{nom}\}$ and $\{\overline{\lambda_1}, \chi\}$. Classification accuracies are given in the second row. (Reproduced from [Publication-II].)

	$\{\chi, d_{nom}\}$ $A = 70.7\%$						$\{\overline{\lambda_1}, d_{nom}\}$ $A = 72.7\%$						$\{\overline{\lambda_1}, \chi\}$ $A = 92.7\%$					
	S1	S2	S3	L1	L2	L3	S1	S2	S3	L1	L2	L3	S1	S2	S3	L1	L2	L3
S1	18	7	0	0	0	0	21	0	0	4	0	0	24	1	0	0	0	0
S2	2	23	0	0	0	0	0	24	0	1	0	0	0	25	0	0	0	0
S3	0	0	24	0	1	0	0	0	22	0	1	2	0	0	25	0	0	0
L1	0	0	0	0	25	0	3	2	0	20	0	0	0	0	0	25	0	0
L2	1	1	2	1	17	3	1	19	0	0	1	4	1	1	0	3	15	5
L3	0	0	0	1	0	24	0	3	1	0	0	21	0	0	0	0	0	25

4.1.3 Performance test

We used binary images generated by thresholding Gaussian random fields (GRF) to test the speed of our implementation. These images are commonly used for studying microstructures [54], they are easy to generate and have a single tuning parameter for controlling the level of details.

We generated GRFs in Fourier domain by using random phases and Rayleigh distributed amplitudes. Rayleigh density function is $\mathcal{R}_{pdf}(x|\sigma_R) = \frac{x}{\sigma_R^2} e^{\frac{x}{2\sigma_R^2}}$ with σ_R being the Rayleigh scaling parameter. We generated 20 samples of dimensions 256×256 for each $\sigma_R = \{4, 7, 10, 13\}$. GRFs with larger σ_R give structurally more detailed images. Sample images obtained for each σ_R is given in Fig.4.9.

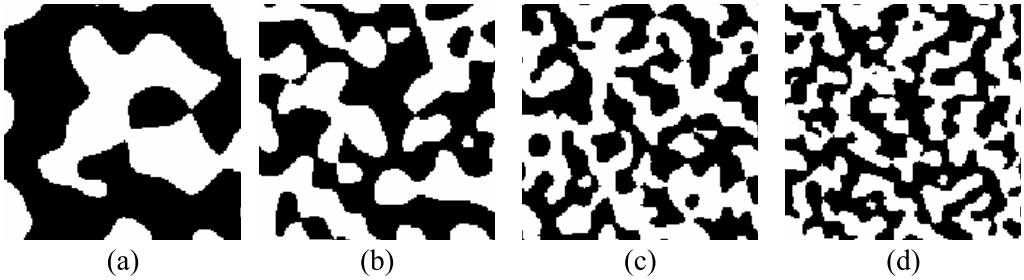


Figure 4.9: Sample test images used in the performance test. Images are generated by thresholding GRFs. Rayleigh scaling parameter, σ_R, is adjusted to vary the level of details. The images correspond to the GRFs generated by (a) $\sigma_R = 4$ (b) $\sigma_R = 7$ (c) $\sigma_R = 10$ and (d) $\sigma_R = 13$.

Table 4.3 shows the mean results for each σ_R group. Here, the computation times for sweeping, merging, pruning, supplementing and eigen value calculation are given separately. Also, the total number of vertices for the trees are given on the last three columns. All computations are done using an AMD Phenom® II X4 955 processor with 12GB system memory.

Table 4.3: Performance test results on GRF images. The values are the averages obtained using 20 images for each $\sigma_R = \{4, 7, 10, 13\}$. The second column shows the average $\overline{\lambda_1}$ which is decreasing with the increasing level of detail.

		Time (s)						Number of vertices		
σ_R	$\overline{\lambda_1}$	Total	Sweep	Merge	Prune	Supplement	Eigen value	CT	PCT	SCT
4	0.152	3.1	0.2	1.9	0.4	0.01	0.6	632	148	690
7	0.079	12.8	0.2	10.0	1.0	0.01	1.5	1328	331	978
10	0.063	34.6	0.2	29.5	2.0	0.02	2.9	2171	512	1219
13	0.053	68.9	0.2	60.5	3.5	0.02	4.7	3137	651	1399

4.2 Case B: Contour tree connectivity and mechanical strength of trabecular bone microstructure

Connectivity has long been recognized as an important parameter to predict the mechanical properties of trabecular bone [44]. The most commonly used connectivity measure is the connectivity density (Conn.D), which is based on the Euler number (χ) [80]. As explained in subsection 2.4.1, Conn.D gives the density of trabeculae in unit volume. However, although Conn.D includes information regarding the number of trabeculae, it does not provide information about the thickness of the connections nor the gaps between them.

In this study, we propose to use the contour tree connectivity (CTC) as an alternative connectivity measure to predict the strength of trabecular bone. In [Publication-I], contour trees are utilized to provide graph representations of binary images. This representation of images transforms complex 3D data, such as the trabecular bone, into simple graphs. Based on a similar representation, CTC calculates the connectivity of binary images by calculating the connectivity of graphs [Publication-II].

In the first subsection we conduct a study using synthetic data in order to show the complementary information provided by CTC. In the second subsection we show the results obtained using patient data.

4.2.1 Tests with synthetic data

4.2.1.1 Description of data

We designed synthetic samples using MATLAB in order to study the effect of number of trabeculae, number of plate-like structures and bone deterioration on bone volume ratio (BV/TV), connectivity density (Conn.D) and contour tree connectivity (CTC). Samples from the computer generated data are shown in Fig.4.10.

The data are generated by a simple arrangement of intersecting cylinders with radius of 8 voxels. The first structural parameter we modified is the number of trabeculae which we altered by changing the density of cylinders. The three groups with different numbers of trabeculae are denoted with Roman numerals I, II, III. The second structural parameter that is modified is the number of plates between the trabeculae. For each groups of I, II and III, we created three different sets of data with varying plate numbers. With the

increasing order of plates, these are denoted with A, B and C. The last parameter that is controlled is bone deterioration, which is simulated by morphological erosion. Eroded bones are denoted with A', B' and C'. In order to account for the changes in random arrangements, we created 20 samples in each group. Therefore, 360 synthetic images with 18 structurally different groups are prepared.

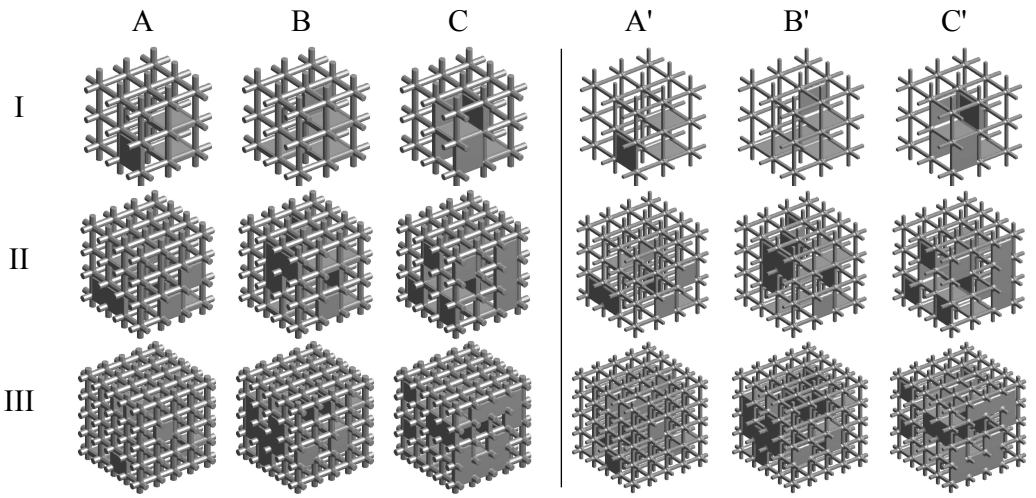


Figure 4.10: Sample images used for the tests with synthetic data. I, II and III denote the groups with different numbers of trabeculae. A, B and C denote the different numbers of plates. A', B' and C' denote that bone is eroded.

4.2.1.2 BV/TV, Conn.D and CTC values

We computed BV/TV, Conn.D and CTC values belonging to each synthetic data group. Fig.4.11 shows the average results, the error bars show the min and max values of each parameter. Bottom row shows the values for the deteriorated bone. Based on the results, the effects of the morphological changes in the parameters can be summarized with the Table-4.4

Table 4.4: The effect of the morphological changes in BV/TV, Conn.D and CTC. Neither Conn.D nor CTC provides information that is correlated with BV/TV. CTC may increase or decrease with the change in the number of plates.

	BV/TV	Conn.D	CTC
↑ number of trabaculae	↑	↑	↓
↑ number of plates	↑	↓	↑ or ↓
↑ bone erosion	↓	no change	↑

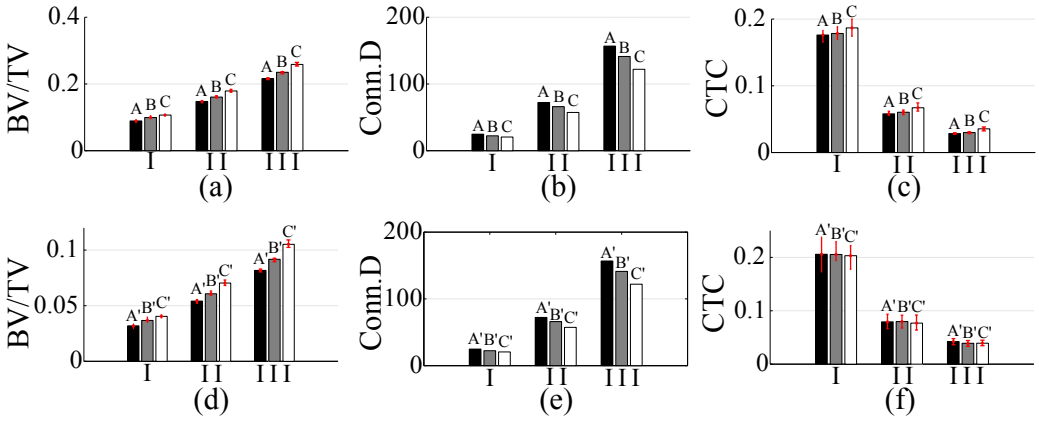


Figure 4.11: Computed BV/TV, Conn.D and CTC values for each group. Bottom row shows the values for the deteriorated bone. Error bars show the min and max values of the parameters. There is no error bar for Conn.D since there is no variation.

It is observed that neither Conn.D nor CTC provides information that is correlated with BV/TV. Given that topology is preserved, Conn.D does not provide information in the case of erosion whereas CTC increases.

4.2.2 Tests with patient data

4.2.2.1 Patient data, μCT imaging and biomechanical testing

The original study population consisted of 61 patients with primary hip osteoarthritis who underwent primary cementless total hip arthroplasty (THA). The demographic data of the patients was described previously in [66]. During THA surgery, a cancellous biopsy was taken from the intertrochanteric region of the proximal femur as described earlier in [76]. The biopsy specimen were wrapped in saline-wet sponges and placed in a sealed plastic bag and stored frozen at -20°C until machining. Multiple specimens were prepared from each biopsy under saline irrigation using a high-speed trephine drill. The cylindrical specimens were scanned with μCT (SkyScan 1072, Kontich, Belgium) with isotropic voxel resolution of 16.3 μm.

From the μCT images, 10 parameters were calculated. These are: bone volume fraction (BV/TV), bone mineral density (BMD), mean trabecular thickness (Tb.Th), trabecular separation (Tb.Sp), trabecular number (Tb.N), trabecular bone pattern factor (Tb.Pf), structure model index (SMI), degree of anisotropy (DA), Conn.D and CTC.

Conn.D and CTC were calculated using MATLAB, whereas the other parameters were calculated using CTan software (SkyScan). For each patient, the mean results of the parameters obtained from multiple specimens were used in the study.

Following the μCT imaging, the specimens were used in compression testing to failure using a universal mechanical testing device (Avalon Technologies, Rochester, MI, USA). Tests were performed at a constant speed of 0.825 mm/min and the load data were continuously recorded by the data acquisition system (Visual Designer, Intelligent Instrumentation, Tucson, AZ, USA). Values for the ultimate tensile strength (σ_U) were calculated from the raw data files using Origin software (Origin Lab Corp. Northampton, MA, USA).

The analyses were performed on 55 patients out of 61. Six patients were excluded due to incomplete data. The patients were divided into three groups with respect to their trabecular bones' ultimate strength. The first group, G-1, has $\sigma_U \leq 0.35$ MPa, the second group, G-2, has 0.35 MPa $< \sigma_U \leq 1$ MPa and the last group, G-3, has $\sigma_U > 1$ MPa. The thresholds were chosen manually by taking into account the jumps in σ_U. According to this division, 26 patients belong to G-1, 20 patients belong to G-2 and 9 patients belong to G-3. Fig.4.12 shows the values for the ultimate tensile strength in ascending order and the threshold levels used to group the patients.

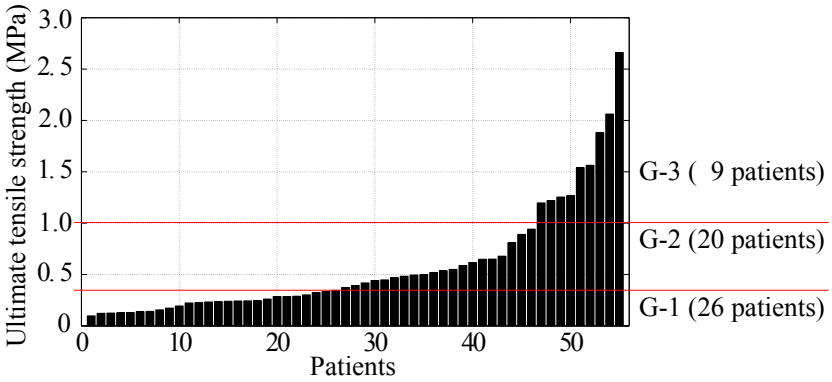


Figure 4.12: Ultimate strength values obtained from the measurements in ascending order. Red lines show the thresholds that are used to categorize the patients into three groups. (Reproduced from [Publication-III].)

4.2.2.2 Regression, feature selection and classification method

The correlation between ultimate tensile strength and BV/TV was reported earlier in [76]. In this work, we study the other correlations and calculate the coefficient of determination (R^2) between *(a)* σ_U and μCT parameters *(b)* BV/TV and μCT parameters *(c)* CTC and μCT parameters.

To identify the most predictive μCT parameters for ultimate tensile strength, we used the data from all the patients and did a sequential forward feature selection (SFS). SFS selects a subset of μCT features that best predict the σ_U measurements by sequentially adding features. Each feature is selected after a repeated application of 10-fold cross validation. Among all candidates, the feature set that gives the minimum mean deviance is picked. Deviance measures the quality of the fit as the residual sum of squares obtained from generalized linear model regression. We used SFS on two sets of features. The first feature set, Case-1, had all the 10 μCT parameters. The second feature set, Case-2, did not have CTC but had all the other 9 parameters.

Lastly, we used linear discriminant analysis for classifications. For this, we used 13, 10 and 4 patients from groups G-1, G-2 and G-3 respectively as the training set. The rest of the 28 patients are used in the tests. For both feature sets, Case-1 and Case-2, we sequentially classified test subjects into groups by increasing the number of significant features.

4.2.2.3 Demonstrative CTC results

We picked three μCT images to show example CTC values for different cases. Fig.4.13 shows these samples, corresponding ultimate tensile strength, BV/TV and CTC values. Since the actual contour trees contain several hundreds of vertices, simplified versions obtained from the crops are plotted below each μCT image.

Fig.4.13 shows example CTC values obtained for 3 μCT images. Generally an increased number of trabeculae decreases CTC by two mechanisms. Firstly, trabeculae appear as local maxima in the EDT image, which introduces vertices into the join tree. Secondly, gaps between trabeculae, that appear as local minima, introduce vertices into the split tree. As a result, a sample with a large number of trabeculae is represented with a spread of several vertices above and below 0 level. In the contour trees shown in

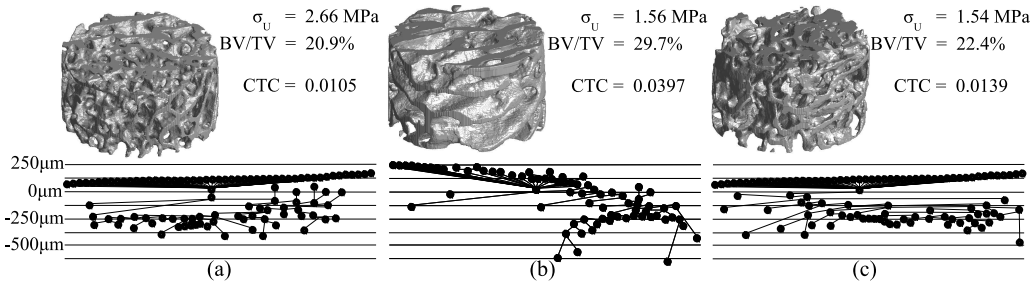


Figure 4.13: Example μCT images and corresponding σ_U, BV/TV and CTC values. Below each sample is a simplified version of the contour tree. (a) shows the strongest bone in the dataset, (b) has the highest BV/TV in the dataset, (c) has a similar σ_U value to the one in (b) despite less BV/TV. (Reproduced from [Publication-III].)

Fig.4.13, vertices at higher levels correspond to thicker trabeculae, whereas lower levels correspond to larger gaps between trabeculae. By quantifying the connectivity of this representation, CTC quantifies the connectivity of trabeculae with regard to their thickness and separation.

Fig.4.13 also shows how CTC complements other features such as BV/TV. From Fig.4.13a to Fig.4.13b, there is a 1.1 MPa decrease in σ_U despite the 8.8% increase in BV/TV. Additionally, from Fig.4.13b to Fig.4.13c, the 7.3% decrease in BV/TV does not change σ_U significantly. The contour tree for Fig.4.13b with few trabeculae and large gaps accounts for these differences. This compact tree representation yields a large CTC value, which might indicate structural weakness for this sample.

4.2.2.4 Results of the correlation analysis

Fig.4.14 shows the calculated R^2 values. From left to right, Fig.4.14 shows the linear correlation between (a) σ_U and μCT parameters (b) BV/TV and μCT parameters (c) CTC and μCT parameters.

The correlation analysis in Fig.4.14a shows that BV/TV alone explains more than 65% of the variance in σ_U. Second most correlated feature with σ_U is Tb.N. The least correlated features are DA, BMD and CTC. Fig.4.14b shows that, Tb.N, Tb.Sp, SMI and Tb.Pf are correlated with BV/TV. On the other hand, CTC is uncorrelated with most parameters.

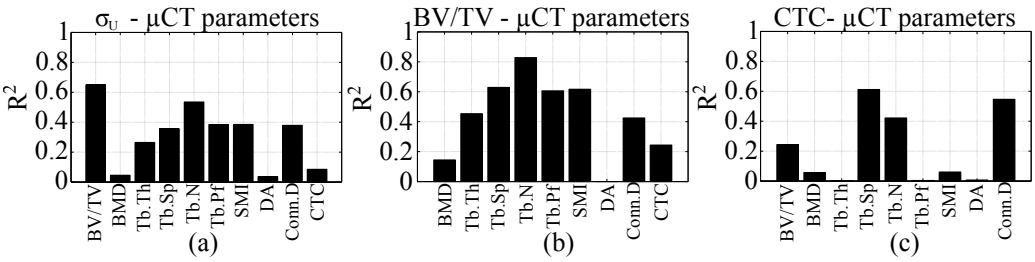


Figure 4.14: Linear regression analyses show (a) σ_U is best correlated with BV/TV (b) Tb.N, Tb.Sp, SMI and Tb.Pf are correlated with BV/TV (c) CTC is uncorrelated with μCT parameters other than Tb.Sp, Conn.D and Tb.N. (Reproduced from [Publication-III].)

4.2.2.5 Feature selection and classification results

Table 4.5 shows the order of the selected features for Case-1 and Case-2 using SFS and the classification accuracies.

Table 4.5: The order of the selected features that best predict σ_U measurements and the classification accuracies using feature sets Case-1 and Case-2. CTC is chosen as the 3^{rd} best feature by SFS. BV/TV and DA are the top two features. (Reproduced from [Publication-III].)

	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
Case-1	BV/TV	DA	CTC	Conn.D	Tb.Sp	Tb.Pf	Tb.Th	Tb.N	SMI	BMD
	60.7%	50%	64.3%	67.9%	67.9%	67.9%	71.4%	71.4%	71.4%	67.9%
Case-2	BV/TV	DA	SMI	Conn. D	Tb.N	Tb.Th	Tb.Sp	BMD	Tb.Pf	-
	60.7%	50%	60.7%	60.7%	60.7%	53.6%	53.6%	53.6%	57.1%	-

Table 4.5 shows that CTC is the 3^{rd} best predictive feature of ultimate strength after BV/TV and DA. This coincides with recent studies where BV/TV and DA are shown to be good predictors for σ_U [118]. Interestingly, our results show that BMD is neither correlated with σ_U nor it is significant when predicting σ_U. It is observed from the correlation and classification results that most of the commonly used parameters are redundant. For Case-2, once BV/TV is used, the inclusion of many features does not improve accuracy. Moreover, additional features decrease the accuracy more for Case-2 than Case-1 due to the curse of dimensionality. Lastly, it is observed that 3 subsets of Case-1 that include CTC reach an accuracy of 71.4%, which could not be achieved by any Case-2 subsets.

4.3 Case C: Contour tree connectivity and fluid flow in microstructures

In this study, we assess and compare the capability of various geometrical characteristics including CTC in relating structural properties with permeability of microstructures. To achieve this goal, we generated 120 samples representing 6 different types of microstructures with equal porosities using MATLAB. We correlated the nominal opening dimension (d_{nom}), CTC, Euler parameter (EP), parameter for connected pairs (PCP) and the permeabilities estimated using direct pore scale modeling. We show that the connectivity interpretation and information provided by CTC is different from the other measures and can be used to quantify connectivity in order to identify differences and changes in microstructures [Publication-IV].

4.3.1 Description of materials and estimation of permeabilities

We prepared random porous structures that resemble segmented microstructures obtained from a μCT device with an isotropic voxel spacing of 7.8125 μm. All samples were designed to have physical dimensions of $1mm \times 1mm \times 1mm$ which correspond to 128×128×128 voxels in image domain. In order to quantify connectivity independently of porosity, all samples were designed to have ∼85% porosity. Porosity was calculated as the ratio of the number of permeable voxels to the number of all voxels.

We generated 6 groups of data where each group contains 20 samples. 3 of the groups contain samples made of spheres and the rest of the 3 contain samples of fibers. We abbreviate the sphere samples as S1, S2, S3 and fiber samples as F1, F2, F3. S1 and F1 have 45 big disconnected objects, S2 and F2 have 100 medium sized disconnected objects and S3 and F3 have 300 small disconnected objects.

For S1, S2 and S3 the diameters of the spheres vary between 15-26, 9-20 and 3-14 voxels respectively. All fibers in F1, F2 and F3 are square prisms with a varying ratio of width to height between 10-20. The widths of fibers in F1, F2 and F3 vary between 6-8, 4-6 and 2-4 voxels respectively. Fig.4.15 shows sample images from each group.

We designed and generated the images using an in-house developed MATLAB software. Starting with an empty image, our code iteratively places a sphere or a fiber of

desired dimensions in a random fashion where ever there is no collision. The iteration stops when the necessary number of objects are placed in the image.

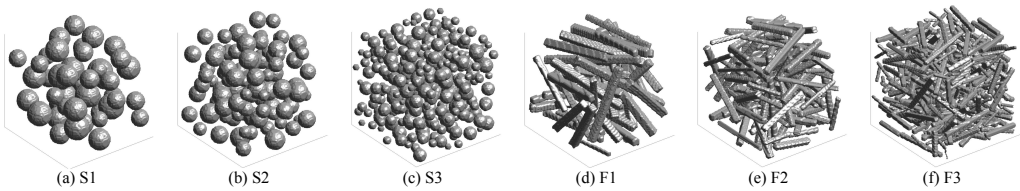


Figure 4.15: Sample visualizations of the 6 type of 3D structures. (Reproduced from [Publication-IV].)

To estimate the permeability of the materials we used direct pore scale modeling (DPLS) or pore-level numerical simulation which has become a popular method to estimate permeability during the last few decades due to its speed and low cost [89]. The common approach used in DPLS is based on the setup shown in Fig.4.16 and Darcy's law, which states that:

$$\nabla p = -\frac{\mu}{K} u_D \tag{4.4}$$

here ∇ is the *del* operator, p is the pressure (N/m^2), μ is the dynamic viscosity (kg/ms) of the fluid, K is the permeability (m^2) of the porous media and u_D is the Darcian velocity (m/s) of the fluid.

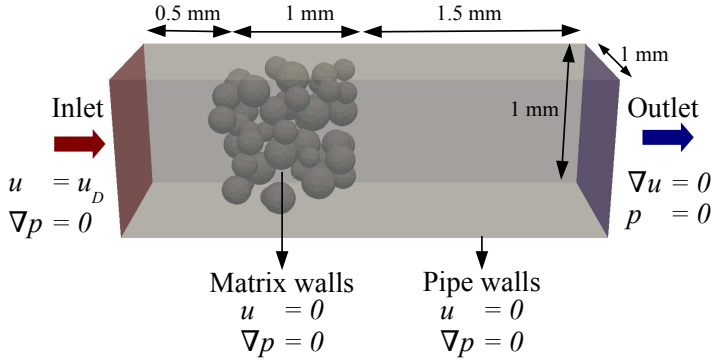


Figure 4.16: The setup used in direct pore scale modeling. The figure shows the initial and boundary conditions for velocity and pressure. (Reproduced from [Publication-IV].)

The main goal of DPLS is to calculate the pressure distribution inside the porous medium for a given velocity. Therefore, the model requires an input u_D, in addition to

the fluid parameters such as μ. The value for u_D is determined based on an assumption on the Reynold's number (Re). Since Equation 4.4 is valid only for low fluid velocities, turbulence is neglected and Reynold's number is assumed to be < 2. Followed by this assumption, u_D can be estimated from the modified expression for the Reynolds number:

$$Re = -\rho \frac{u_D}{\varepsilon} \frac{d_{nom}}{\mu} \Longrightarrow u_D = -\frac{\varepsilon \mu Re}{d_{nom}\rho} \tag{4.5}$$

here ρ is the density of the fluid (kg/m^3), ε is the porosity and d_{nom} is the nominal opening dimension (m) of the medium.

Lastly, by combining Equations 4.4 and 4.5, K can be expressed as:

$$K = \frac{\varepsilon \mu^2 Re}{d_{nom}\rho \nabla p} \tag{4.6}$$

here the only unknown is the pressure gradient, which can be estimated by a Navier-Stokes solver. The governing equations are:

$$\begin{aligned} \nabla.\mathbf{v} &= 0 \\ \rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) - \mu\nabla^2\mathbf{v} &= -\nabla p \end{aligned} \tag{4.7}$$

where $\mathbf{v}$ is the velocity vector. Equation 4.7 couples velocity and pressure and can be solved by the SIMPLE algorithm (**S**emi **I**mplicit **M**ethod for **P**ressure **L**inked **E**quation), which is a popular approach applied to obtain a steady state solution for incompressible fluid flow [88].

As a Newtonian, incompressible fluid, we use water in the models. Specifically, we use the fluid properties at 20°C with a density of $\rho = 998.2\ kg/m^3$ and a dynamic viscosity of $\mu = 0.001\ kg/ms$ [62]. In the models, water is flowed through a rectangular pipe of dimensions $1mm \times 1mm \times 3mm$. This setup is shown in Fig.4.16. As boundary conditions, 0 velocity on the sides of the tube and 0 pressure on the outlet is used. For the walls of the porous medium, no slip condition is assumed. Turbulence is neglected and a laminar flow with $Re = 0.12$ is modeled. The initial condition for u_D at the inlet is calculated using Equation 4.5. Time step for the simulation is determined by setting the Courant number to 0.1. OpenFoam and SIMPLE algorithms are used for the solution of the pressure distribution.

The meshing of the geometry is done using the enGrid software [1]. Fig.4.17 shows visualizations for the meshes of a sample belonging to group S3. Inlet and outlet for water are shown with burgundy and purple respectively. The dimensions of the tetrahedral elements are adjusted by the software. The example mesh in Fig.4.17 contains 1.3e6 million tetrahedral elements which is set in consistence with the predetermined Courant number.

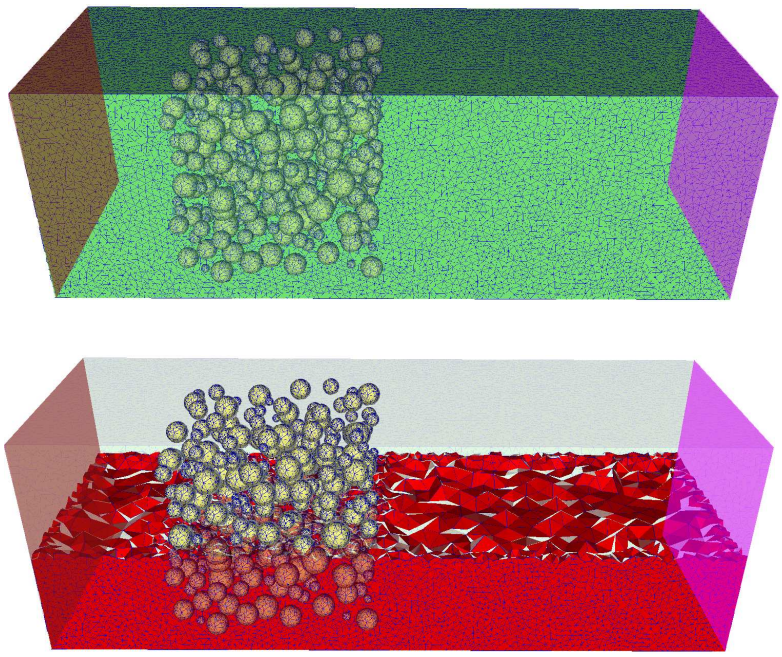


Figure 4.17: Visualizations of an example mesh belonging to group S3. Inlet and outlet for water are shown with burgundy and purple respectively. Surface visualization on the left contains 1.73e3 triangular elements. Green triangles show the model boundary. The walls of the porous medium is shown with yellow. Volumetric visualization on the right contains 1.3e6 tetrahedral elements. Red tetrahedrons mark the region where water flows.

4.3.2 Contour trees and CTC values

From each group of test materials, we picked a sample to show example contour trees and the corresponding CTC values. The samples are shown in Fig.4.15. The corresponding contour trees and the calculated CTC values are shown in Fig.4.18. Samples with a fewer number of objects (S1 and F1) have less vertices in their contour trees due to a fewer number of local maxima and minima in their EDT image (not shown). With the increase in the number of objects, the number of local maxima and minima also increase. As a result, samples with more objects contain more vertices in their contour trees (S3 and F3).

Increasing the number of pendant vertices decreases algebraic connectivity. Therefore, in general, increasing the number of objects in a random fashion decreases CTC.

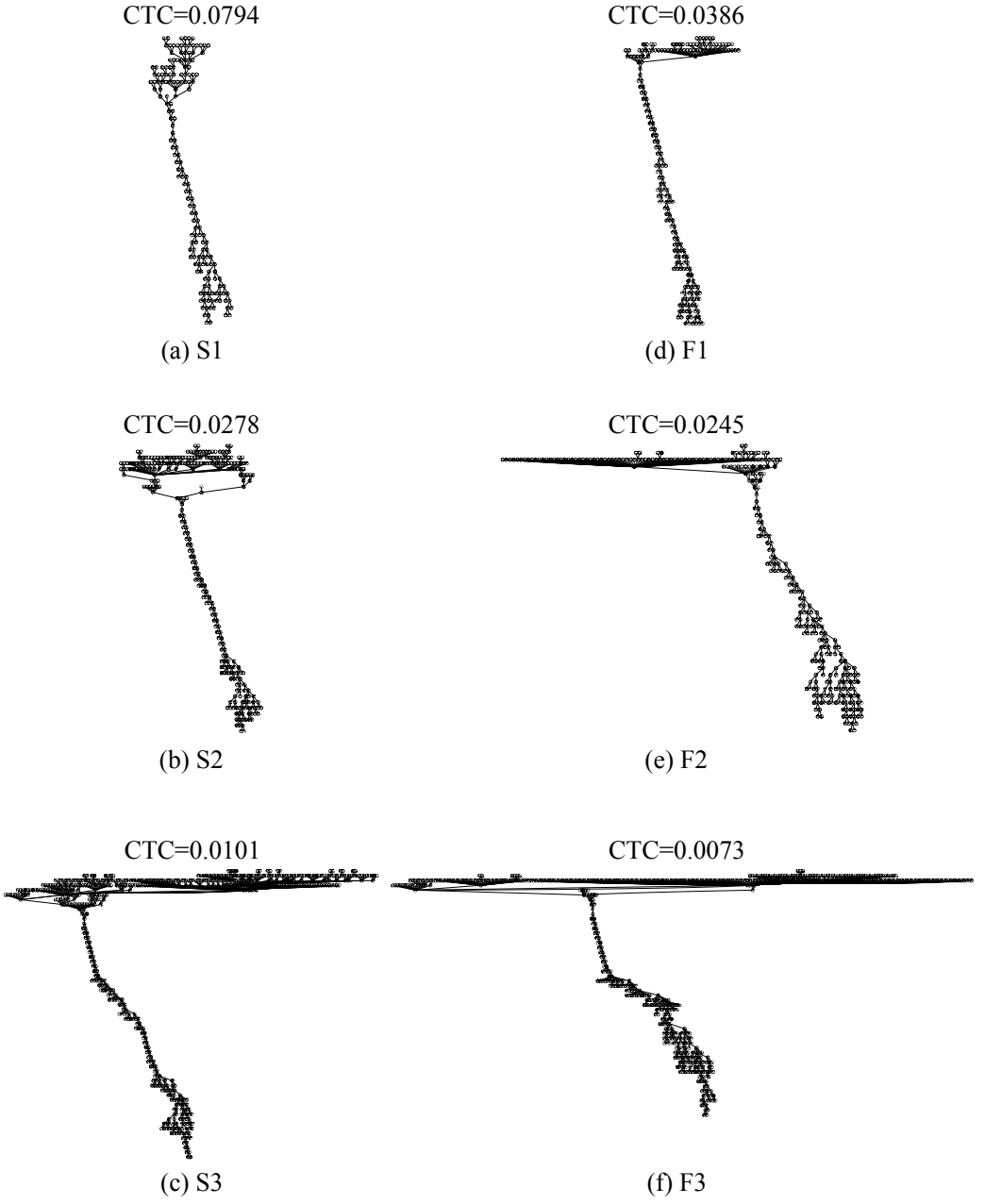


Figure 4.18: Example contour trees and CTC values. The trees belong to the samples shown in Fig.4.15. Increasing the number of objects introduces more details in the graph which decreases CTC. (Reproduced from [Publication-IV].)

4.3.3 Permeability, d_{nom}, CTC, Euler parameter (EP), Parameter for connected pairs (PCP)

Fig.4.19a shows an example of the velocity distribution through the material and the resulting pressure distribution is shown in Fig.4.19b. A negative pressure gradient between 0.5mm-1.5mm is shown in Fig.4.19c where the material is placed. This trend is observed for all the samples. By using Equation 4.6 and the obtained pressure gradients, permeabilities are estimated. Examples values are shown in Fig.4.19c.

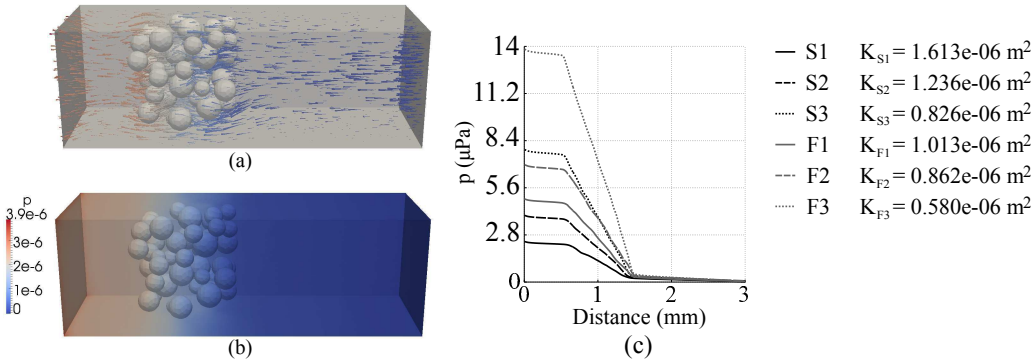


Figure 4.19: (a) Velocity distribution and (b) pressure distribution computed for the S1 sample shown in Fig.4.15a (c) Example pressure gradients and permeability values for samples belonging to different groups. (Reproduced from [Publication-IV].)

The calculated permeability, d_{nom}, CTC, EP and PCP values are plotted for each test group in Fig.4.20. Table 4.6 shows the means and variances of the parameters for each group. EP is the mean of the absolute value of the Euler numbers obtained for the excursion sets which was explained in section 2.4.1. Similarly, PCP is the mean of the Γ curve computed on the excursion sets which was explained in section 2.4.2.

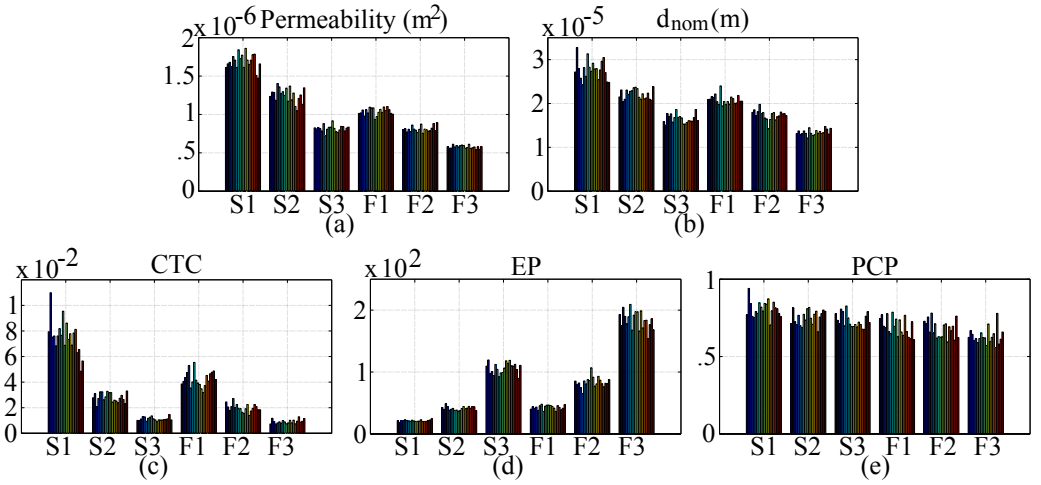


Figure 4.20: FE and image analyses results for all the test samples (a) permeability (b) d_{nom} (c) CTC (d) EP (e) PCP. For each group different colors represent different samples. (Reproduced from [Publication-IV].)

From Table 4.6, it is observed that among the test groups, S1 has the highest mean permeability and is followed by S2 and F1. The lowest mean permeability value belongs to the group F3. F2 and S3 also have low permeability values. A similar trend is observed with d_{nom} and EP (inversely). However, CTC and PCP do not show this trend.

Table 4.6: Mean (μ) and variances (σ^2) of the parameters calculated for each group. (Reproduced from [Publication-IV].)

		S1	S2	S3	F1	F2	F3
Permeability (m^2)	μ	1.69e-6	1.25e-6	0.82e-6	1.04e-6	0.82e-6	0.58e-6
	σ^2	9.90e-15	8.69e-15	1.67e-15	2.08e-15	1.49e-15	0.39e-15
d_{nom} (m)	μ	2.77e-5	2.21e-5	1.65e-5	2.08e-5	1.74e-5	1.34e-5
	σ^2	4.88e-12	1.25e-12	1.05e-12	1.11e-12	1.22e-12	0.41e-12
CTC	μ	7.54e-2	2.81e-2	1.13e-2	4.24e-2	1.99e-2	0.92e-2
	σ^2	1.73e-4	0.13e-4	0.02e-4	0.37e-4	0.09e-4	0.02e-4
EP	μ	21.24	41.28	106.13	42.98	83.62	184.06
	σ^2	1.82	9.02	78.48	12.60	68.52	205.39
PCP	μ	0.81	0.75	0.73	0.69	0.68	0.63
	σ^2	2.70e-3	2.14e-3	2.03e-3	3.45e-3	3.06e-3	2.48e-3

4.3.4 Correlation, clustering analysis and classification results

Fig.4.21 shows scatter plots between permeability, d_{nom} and CTC, EP, PCP. Table 4.7 shows the values for the coefficients of determination (R^2). The first column shows the R^2 values calculated using all the test samples and the rest of the columns show values corresponding to each group separately.

From Fig.4.21a-c and Table 4.7, it is observed that d_{nom} has the highest linear correlation with permeability ($R^2 = 0.91$). CTC, EP and PCP follows d_{nom} with R^2 values of 0.78, 0.7 and 0.45 respectively.

Table 4.7: The coefficients of determination (R^2) obtained using linear correlation analysis. The first column shows the R^2 values calculated using all the test samples. Other columns show the correlation between the parameter pairs for each group separately. (Reproduced from [Publication-IV].)

	All	S1	S2	S3	F1	F2	F3
d_{nom}-Permeability	0.91	0.17	0.22	0.00	0.02	0.00	0.04
CTC-Permeability	0.78	0.08	0.14	0.03	0.08	0.01	0.25
EP-Permeability	0.70	0.07	0.20	0.00	0.16	0.06	0.13
PCP-Permeability	0.45	0.07	0.03	0.29	0.00	0.24	0.18
CTC-d_{nom}	0.84	0.40	0.70	0.66	0.47	0.61	0.51
EP-d_{nom}	0.78	0.65	0.82	0.92	0.60	0.88	0.85
PCP-d_{nom}	0.54	0.69	0.16	0.20	0.21	0.42	0.31

In order to quantify how well the groups are separated in Fig.4.21, we calculated the distances between the groups using the Bhattacharyya distance. Larger Bhattacharyya distance indicates better separation of groups. Fig.4.22 shows the logarithm of the distances between groups. It is observed that although d_{nom} is the best predictor for permeability, d_{nom}-permeability combination is not able to separate S2 from F1 and S3 from F2. These groups are also not discriminated well with EP-permeability and PCP-permeability combinations. CTC-permeability combination gives the largest distance between the similar groups. Using d_{nom} instead of permeability as the second feature decreases almost all the distances between groups. However, CTC-d_{nom} combination still provides a good contrast between groups. EP-d_{nom} combination is better at discriminating S3 and F2 compared to EP-permeability, however, S2 and F1 are still not separated well. PCP-d_{nom} combination does not provide useful information to separate different groups.

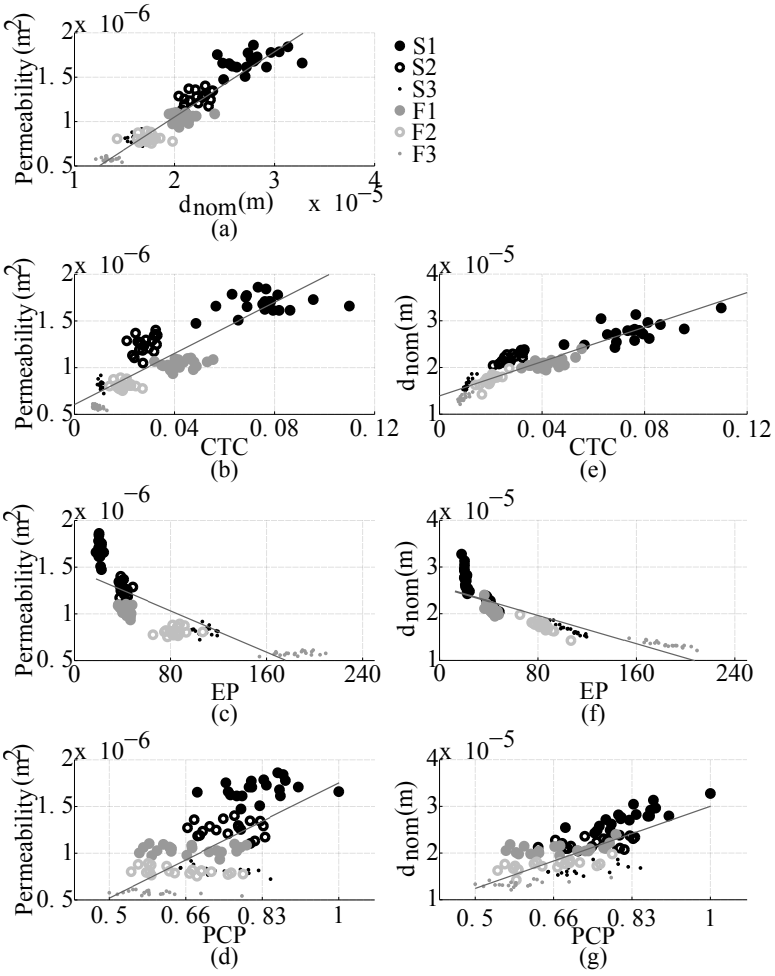


Figure 4.21: Scatter plots of different parameter combinations. Straight lines shown in the plots show the best linear fits calculated using all the samples. R^2 values corresponding to the fits are given in Table 4.7. (Reproduced from [Publication-IV].)

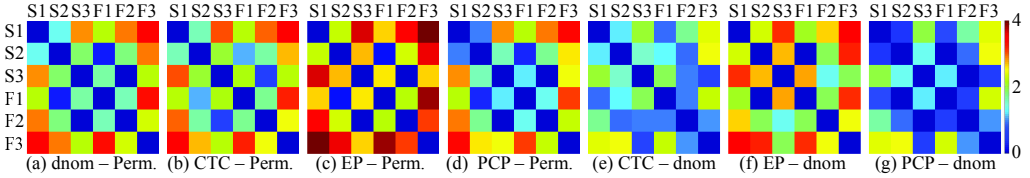


Figure 4.22: Logarithm of the Bhattacharyya distances between groups using different parameter combinations. Largest values of separation are obtained using EP. However both EP-permeability and EP-d_{nom} feature combinations fail to separate S2 and F1. CTC provides a good contrast between groups that are difficult to be separated by other parameter combinations. (Reproduced from [Publication-IV].)

Lastly, we performed a classification test using the same parameter combinations shown in Fig.4.21 We used a supervised classification approach based on linear discriminant analysis (LDA). Half of the samples from each group were used to train a classifier and the rest of the samples were used for the tests. We repeated the experiments 100 times with different training and test groups. Fig.4.23 shows the average results for the classification tests and accuracies using each parameter combination. Rows in Fig.4.23 show the true classes and columns show the predicted ones.

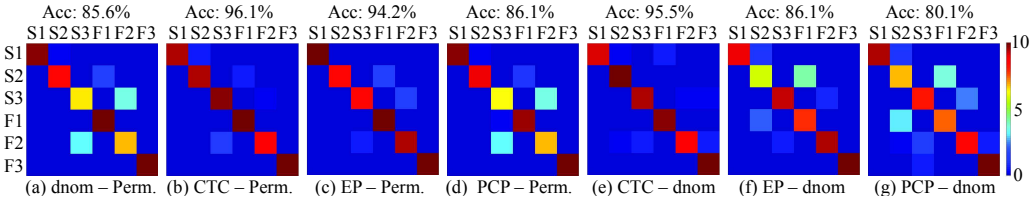


Figure 4.23: Results of the classification tests. Rows of the confusion matrices show the true classes and columns show the predicted ones. Best classification accuracy is obtained using CTC-permeability combination. EP also performs well when used together with permeability. In the absence of permeability information, the best classification result is obtained using CTC-d_{nom} combination. (Reproduced from [Publication-IV].)

The value of CTC varies due to several factors affecting the geometry and topology, such as the amount of connected components, their proximity to each other, cavities or holes they have. The example contour trees shown in Fig.4.18 demonstrate that in general compact graphs give larger CTC values and details lower CTC. Although, the information CTC provides is important to quantify structural properties, we are also interested in correlating this information with permeability.

Table 4.7 and the scatter plots show the correlation between parameters. d_{nom} is the most correlated parameter with permeability ($R^2 = 0.91$). The scatter plots in Fig.4.21e-g show the relations between d_{nom}, CTC, EP and PCP. This demonstrates the case where costly FE modeling or experiments are not possible and permeability values are not known. Because d_{nom} is highly correlated with permeability, we investigated its relation with other parameters. Relations between CTC, EP and PCP are not shown since they did not yield much information about flow and microstructures.

In order to visually explain how each parameter separates different groups, we plotted the Bhattacharyya distances in Fig.4.22. It is observed that although d_{nom} is the most correlated parameter with permeability, it does not provide as useful information as CTC

in terms separating structural differences. For example, S3 and F2 have similar permeability and d_{nom} but with different structures. On the other hand, CTC is both correlated with permeability ($R^2 = 0.78$) and can separate differences in structures. Both EP and PCP has lower correlations with permeability, additionally, they both fail to separate the structural differences within groups.

5 DISCUSSIONS

With the goal of providing a new connectivity interpretation, quantifying it with a scalar number and investigating how this measure is linked to real-life processes, we conducted experiments in Chapter 4 and tried to show how many of these goals were met. Indeed, by using contour tree representations of binary images as shown in [Publication-I] and [Publication-II], contour tree connectivity (CTC) uses a different interpretation of structural connectivity. The geometric interpretation of connectivity with CTC, complements the topological connectivity information given by the Euler number (χ).

There are several other related graphs to contour trees that are used to represent images. For example, the *Morse-Smale complex* and *Reeb graphs* are closely related tools used in computational topology [9]. Unlike contour trees, Reeb graphs may have cycles. However, if a Reeb graph of a scalar field is a tree then it is identical to the contour tree of the scalar field. Due to this relation, contour trees and Reeb graphs can be computed using similar algorithms [90]. In image processing, join and split trees are also known as *max-* and *min-trees* [97]. The *level line tree* or the *tree of shapes* is a joint representation of max- and min-trees [74] that has similarities to contour trees. A *concavity tree* is another structure used for describing non-convex 2D shapes [108]. In contrast to concavity trees, which are developed for 2D, contour trees are applicable to all dimensions. Additionally, contour trees represent scalar fields based on topological properties unlike max-, min- and the level line trees that put on hierarchical representations of level-sets or shapes. Also

when compared to Morse-Smale complex and Reeb graphs, contour trees are easier to work with since they are trees. Due to these reasons, we picked contour trees to quantify the connectivity of structures.

We introduced CTC in [Publication-II] and its applications on quantifying the connectivity of microstructures in [Publication-III] and [Publication-IV]. One of the challenges we faced was to balance the new theory and applications in our articles. We published most of the results shown in section 4.1 in [Publication-II] that aimed to introduce CTC with all the necessary theoretical content. Since in [Publication-II], we focused on showing how and why CTC works, we worked with the 2D toy images shown in Fig.4.1 and artificial the 3D images shown in Fig.4.6. The results using the toy images show the various properties of CTC. For example, CTC decreases when objects in the foreground separate or when cavities and branches form in the foreground. When foreground objects approach each other or become rounder, however, CTC increases. Although, we only refer to the changes in the foreground, when describing the properties of CTC, it should be kept in mind that CTC is not calculated for the foreground or the background separately. There is only one CTC for a binary scene and it is independent of which binary component is assigned to be the foreground or the background. Therefore, there is actually a duality between foreground and background. This is due to the contour tree representation of binary images in which objects in the foreground are equivalent to holes in the background and cavities in the foreground are equivalent to branches in the background. Furthermore, it should be mentioned that when considering the contour tree representation of binary images, there is little difference between cavities and holes. Actually, this is the main reason why CTC works effectively even on the very noisy images that were studied using the images shown in Fig.4.2. The advantage of avoiding the purely topological connectivity interpretation shows itself when spatial relationships between or within components in the scene vary. It should be noted that the method we use to introduce noise to the binary images also change the topology, which in turn alters χ significantly, as shown in Fig.4.3.

One of the main goals of our work was to design an easy to use scalar measure for connectivity. Thanks to the normalization step that comes from the lower and upper bounds of the algebraic connectivity of supplemented contour trees (SCT), the value of CTC is always between 0 and 1. It is worth mentioning here that normalization is not

done to force a value between 0 and 1. The normalization step is necessary in order to compare the algebraic connectivity of trees with different heights. Another normalization could have been done, for example, in terms of the number of vertices in a tree. This has, however, been shown to be a significantly more difficult problem [65]. Therefore, by normalizing with respect to the height of the trees, it was not only possible to sort trees according to their algebraic connectivity but, as a result, a value between 0 and 1 was also obtained. The supplementation of contour trees, however, introduces additional computation to the algorithm. Furthermore, the size of the Laplacian matrix significantly increases with supplementation and makes the eigenvalue computation slower. With supplementation, geometrical information is introduced to the tree that would not otherwise be there. Another approach to include geometrical information could be the use of edge weights. However, the normalization of weighted trees is again a difficult problem to solve [65, 73]. In short, supplementation together with normalization makes CTC a measure that is between 0 and 1 and is comparable between different images.

Pruning is an essential step that must be performed for CTC. Neither of the two parameters used for pruning (area/volume and level thresholds), however, need careful adjustment. We showed in [Publication-II] that there is a broad confidence zone where these parameters can be adjusted without significantly altering CTC. Relying on this fact, the pruning parameters can be set to constants for practical applications. With this knowledge, we conducted all the experiments in [Publication-III] and [Publication-IV] using constant values.

The advantages of the new connectivity interpretation of CTC come with a cost associated with computational complexity. Most topological invariants such as χ can be easily computed. Even for very large images, there are fast algorithms available [35]. On the other hand, the performance analysis shown in subsection 4.1.3 shows that even for small images, depending on their complexity, computation of CTC can take over a minute. All the results shown in this thesis were obtained by using the same code that is the implementation of the algorithms given in subsection 3.1.4. Our code is written mostly in MATLAB and only the join and split tree computations are done with mex. The performance test results shown in Table 4.3 show that the merging time increases dramatically with the increase in image detail. Although for most binary images our current implementation runs at an acceptable speed, this can be improved with the use of

better implementation and faster programming language. Carr's algorithm to calculate contour trees is $O(n \log n + M\alpha(M))$ for a simplicial mesh of n vertices and M simplices [22]. We used the same approach for merging; we did not, however, use the union-find data structure in our implementation. Therefore, we expect a significant improvement in speed if the code is implemented completely with mex using a union-find data structure.

During the writing of our first publication, [Publication-I], CTC was not mature enough. Thus, we provided only the part where we represented binary images using contour trees, as explained in section 3.2. In this earlier work, we attempted to lower the computational complexity by skipping the ordering step that is done in the very beginning of the sweep tree computation (Algorithm 1, line 2). Instead of sorting, we used all the integer valued levels between the global extrema. This approach indeed increased the computation speed. The speed is increased not only due to skipping the sorting step but also by using a significantly lower number of levels for computation. The resultant contour tree is very much simplified, which eliminates the need for pruning completely. However, the trade off with using predefined levels is the loss of accuracy in the contour tree. Although, we aimed to speed up the contour tree computation by taking advantage of the monotonicity in Euclidean distance transformed (EDT) images, we realized that this reduces the information content in the contour trees. This caused us to reconsider this approach and we decided to implement the algorithms presented in subsection 3.1.4. With the new implementation it was possible to harvest more useful information from the resultant contour trees and this eventually led to CTC.

One of the major applications of CTC is to relate the new connectivity information with real-life processes. To do this, we investigated the correlation between CTC with mechanical strength and the permeability of microstructures. One of the biggest challenges of these studies was the lack of alternative parameters against which CTC could be compared. Other than the Euler number, there is no generally-applicable structural connectivity measure in the literature. However, as shown in [Publication-IV], for flow inside microstructures, percolation analysis is an increasingly used method. Therefore, in addition to the Euler number, we used a proportion of connected pairs for comparison against CTC in [Publication-IV]. It should be pointed out that measurements for connectivity (such as permeability) might not always be possible, as is in the case with measurements

that need to be made on biological tissues in-vivo. The lack of image-based static connectivity measures indicates the need for more research in this area. In both [Publication-III] and [Publication-IV], we showed that the connectivity information provided by CTC is valuable in understanding how the structural properties of microstructures affect physical processes. In particular, [Publication-IV] demonstrates that the Euler number and percolation can be very limited measures for comparing structures with similar topologies. Since CTC is designed to be able to compare such structures, it is reasonable to expect more discriminative connectivity information. This expectation is well met particularly where we sorted the features in order of importance in Table 4.5 that shows that CTC is, after bone volume ratio and anisotropy, the 3^{rd} best predictive feature of the ultimate strength of bone. This coincides with recent studies where these two parameters are shown to be good predictors for the strength of trabecular bone [118].

One major criticism of our studies, in particular for [Publication-IV], is the lack of models for complex structures and the use of real-data. We agree that it is an interesting problem to study CTC on different types of microstructures such as foams, fibers, powders and sintered materials etc. Due to the large variability of microstructures and the complexity associated with each structure, however, there is a need to simplify the problem. Thus, in this thesis we use simple-synthetic structures that are capable of explaining the relationship between morphology and physical processes. In [Publication-IV], we use a simple dataset where differences in structures can be observed using visual inspection. It is, however, extremely difficult to sort the test groups with regard to permeability using visual inspection. This is because of the complicated physics that govern fluid flow. We do agree, however, that using real data would have strengthened our conclusions. On the other hand, because CTC is a new measure, we find it important to study simple data in order to show the reader how and why CTC works. Our future work will indeed involve tests that use real data, but we found it an essential step to first show that CTC is a promising measure on simple synthetic data. This justifies, therefore, the reason and resources to be invested in studying real data.

In this thesis, we propose CTC as a measure for connectivity that can be used to analyze binary images, in particular images of 3D microstructures. CTC is an intriguing measure for connectivity since it can compare the connectivities of scenes, regardless of the differences in topology. We show with both simple synthetic structures and a

real-life application that other alternatives, such as the Euler number and percolation based parameters, do not provide connectivity information that is discriminative enough. Therefore, CTC is also important because of the lack of good alternative measures to quantify connectivity. CTC is not yet, however, fully mature. There is still much to improve, not only with the implementation but also with the theory that might yield faster algorithms for computation. While we are working on improving CTC and its computation speed, we also find it as important to study more real data and to better use CTC for practical applications.

6 CONCLUSION

We aimed to design a scalar connectivity measure that is easy to understand. Such a measure is not only a valuable parameter for classifying and analyzing data but also important for understanding complex structures and real life processes. However, it is quite challenging to develop a robust connectivity index that works well under noisy conditions and is applicable to any topology. With this work, contour tree connectivity (CTC) is proposed to solve the problems associated with complex structures, noise and ease of use.

Based on the results presented in this thesis, the following can be concluded:

- The challenges of working with complex structures can be significantly reduced by using simpler representations. In this thesis, we started with this idea and showed in [Publication-I] that complex structures can be represented with simpler contour trees. We presented results using 2D and 3D synthetic images as well as real trabecular bone data. So far, we used this simple representation only to develop CTC. However, there is clearly more to be investigated from the contour tree representation of binary images.

- We showed in [Publication-II] that the value of CTC is between 0 and 1 which makes it easy to be used. CTC also provides useful information even under very noisy conditions. Moreover, it has a unique interpretation for connectivity and

provides new information which is not available with other methods. Although we applied CTC only to analyze microstructures, it is evident that the information can be used for several other computer vision applications as well.

- In [Publication-III] and [Publication-IV], we showed that CTC is an important parameter to study in order to relate microstructures with physical properties of materials. It is a viable alternative to commonly used parameters such the Euler number and percolation based connectivity measures. It can be used to predict the mechanical strength of microstructures [Publication-III] and analyze how structural properties relate to flow [Publication-IV].

Our approach in developing a new connectivity measure is based on contour trees and algebraic connectivity. Although the histories of both computational topology and spectral graph theory are long, to our knowledge CTC is the first product of these two fields. In addition to the work presented in this thesis, we believe there is much to improve in theory, algorithms, implementation of the method and utilization of CTC for practical applications.

REFERENCES

[1] **enGrid - Open Source Mesh Generation**. http://engits.eu/en/engrid, enGits GmbH Haldenweide 12, 79674 Todtnau, Germany, retrieved April 26, 2013. 65

[2] A. Akolkar and J. Petrasch. **Tomography-Based Characterization and Optimization Of Fluid Flow Through Porous Media**. *Transport in Porous Media*, **95**(3). 13

[3] Denis Allard. **On the Connectivity of Two Random Set Models: The Truncated Gaussian and the Boolean**. In Amilcar Soares, editor, *Geostatistics Tria 92*, number 5 in Quantitative Geology and Geostatistics, pages 467–478. Springer Netherlands, January 1993. 9

[4] Lars Arge and Morten Revsbæk. **I/O-Efficient Contour Tree Simplification**. In Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra, editors, *Algorithms and Computation*, number 5878 in Lecture Notes in Computer Science, pages 1155–1165. Springer Berlin Heidelberg, 2009. 36

[5] R.C. Atwood, J.R. Jones, P.D. Lee, and L.L. Hench. **Analysis of pore interconnectivity in bioactive glass foams using X-ray microtomography**. *Scripta Materialia*, **51**(11):1029–1033, November 2004. 9, 13

[6] M. Aubry, U. Schlickewei, and D. Cremers. **The wave kernel signature: A quantum mechanical approach to shape analysis**. In *2011 IEEE Inter-*

national Conference on Computer Vision Workshops (ICCV Workshops), pages 1626–1633, 2011. 7

[7] C. L. Bajaj, V. Pascucci, and D. R. Schikore. **Seed sets and search structures for optimal isocontour extraction**. Technical report, Texas Institute of Computational and Applied Mathematics, 1999. 16

[8] Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. **The contour spectrum**. In *Proceedings of the 8th conference on Visualization '97*, VIS '97, pages 167–ff., Los Alamitos, CA, USA, 1997. IEEE Computer Society Press. 16

[9] S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo. **Describing Shapes by Geometrical-topological Properties of Real Functions**. *ACM Comput. Surv.*, **40**(4):12:112:87, October 2008. 7, 73

[10] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. **Reeb graphs for shape analysis and applications**. *Theoretical Computer Science*, **392**(13):5–22, February 2008. 7

[11] M. Bojan. **Eigenvalues, Diameter, and Mean Distance in Graphs**. *Graphs and Combinatorics*, **7**:53–64, 1991. 43

[12] F L Bookstein. **Landmark methods for forms without landmarks: morphometrics of group differences in outline shape**. *Medical image analysis*, **1**(3):225–243, April 1997. 7

[13] R. L. Boyell and H. Ruston. **Hybrid techniques for real-time radar simulation**. In *Proceedings of the November 12-14, 1963, fall joint computer conference*, AFIPS '63 (Fall), pages 445–458, New York, NY, USA, 1963. ACM. 16, 26

[14] U. Braga-Neto and J. Goutsias. **A Theoretical Tour of Connectivity in Image Processing and Analysis**. *Journal of Mathematical Imaging and Vision*, **19**(1):5–31, 2003. 5

[15] E. J. Breen and R. Jones. **Attribute openings, thinnings, and granulometries**. *Computer Vision and Image Understanding*, **64**(3):377–389, 1996. 8

[16] P. T. Bremer, G. H. Weber, J. Tierny, V. Pascucci, M. S. Day, and J. B. Bell. **A topological framework for the interactive exploration of large scale turbulent combustion**. In *e-Science 2009 - 5th IEEE International Conference on e-Science*, pages 247–254, 2009. 16

[17] S. Brin. **The anatomy of a large-scale hypertextual Web search engine 1**. *Computer Networks*, **30**(1-7):107–117, 1998. 41

[18] S. R. Broadbent and J. M. Hammersley. **Percolation processes. I. Crystals and Mazes**. *Proceedings of the Cambridge Philosophical Society*, **53**:629–641, July 1957. 11

[19] P. Brard, G. Besson, and S. Gallot. **Embedding Riemannian manifolds by their heat kernel**. *Geometric & Functional Analysis GAFA*, **4**(4):373–398, July 1994. 7

[20] H. Carr. *Topological Manipulation of Isosurfaces*. PhD thesis, Department of Computer Science, The University of British Columbia, 2004. 23, 27

[21] H. Carr and J. Snoeyink. **Path seeds and flexible isosurfaces using topology for exploratory visualization**. In *Proceedings of the symposium on Data visualisation 2003*, VISSYM '03, pages 49–58, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 16, 27, 33, 34

[22] H. Carr, J. Snoeyink, and U. Axen. **Computing contour trees in all dimensions**. In *Proceedings of the eleventh annual symposium on Discrete algorithms*, pages 918–926, San Francisco, California, United States, 2000. Society for Industrial and Applied Mathematic. 7, 15, 17, 27, 28, 29, 43, 76

[23] H. Carr, J. Snoeyink, and M. Van De Panne. **Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree**. *Computational Geometry: Theory and Applications*, **43**(1):42–58, 2010. 16

[24] H. Carr, J. Snoeyink, and Michiel van de Panne. **Simplifying flexible isosurfaces using local geometric measures**. In *IEEE Visualization, 2004*, pages 497–504, 2004. 36, 37

[25] Hamish Carr and Jack Snoeyink. **Representing Interpolant Topology for Contour Tree Computation**. In Hans-Christian Hege, Konrad Polthier, and Gerik Scheuermann, editors, *Topology-Based Methods in Visualization II*, Mathematics and Visualization, pages 59–73. Springer Berlin Heidelberg, 2009. 28

[26] Jun Chen, Chaofei Qiao, and Renliang Zhao. **A Voronoi interior adjacency-based approach for generating a contour tree**. *Computers & Geosciences*, **30**(4):355–367, 2004. 28

[27] Jun Chen, Renliang Zhao, and Zhilin Li. **Voronoi-based k-order neighbour relations for spatial analysis**. *ISPRS Journal of Photogrammetry and Remote Sensing*, **59**(1–2):60–72, 2004. 28

[28] Yi-Jen Chiang, Tobias Lenz, Xiang Lu, and Günter Rote. **Simple and optimal output-sensitive construction of contour trees using monotone paths**. *Computational Geometry*, **30**(2):165–195, February 2005. 28

[29] Ronald R. Coifman and Stphane Lafon. **Diffusion maps**. *Applied and Computational Harmonic Analysis*, **21**(1):5–30, July 2006. 7

[30] T. Cronin. **Automated reasoning with contour maps**. *Computers and Geosciences*, **21**(5):609–618, 1995. 16

[31] D. Cvetković and S. Simić. **Graph spectra in Computer Science**. *Linear Algebra and its Applications*, **434**(6):1545–1562, 2011. 41

[32] Antonio D'Amore, John A. Stella, William R. Wagner, and Michael S. Sacks. **Characterization of the complete fiber network topology of planar fibrous tissues and scaffolds**. *Biomaterials*, **31**(20):5345–5354, July 2010. 1, 8

[33] Mark de Berg and Marc van Kreveld. **Trekking in the Alps without freezing or getting tired**. In Thomas Lengauer, editor, *AlgorithmsESA '93*,

number 726 in Lecture Notes in Computer Science, pages 121–132. Springer Berlin Heidelberg, 1993. 26

[34] M. F. Demirci, R. H. van Leuken, and R. C. Veltkamp. **Indexing through laplacian spectra**. *Computer Vision and Image Understanding*, **110**(3):312–325, 2008. 41

[35] S. Dey, B.B. Bhattacharya, M.K. Kundu, and T. Acharya. **A fast algorithm for computing the Euler number of an image and its VLSI implementation**. In *Thirteenth International Conference on VLSI Design, 2000*, pages 330–335, 2000. 75

[36] M. Diligenti, M. Gori, M. Maggini, and E. Martinelli. **Adaptive graphical pattern recognition for the classification of company logos**. *Pattern Recognition*, **34**(10):2049–2061, 2001. 17

[37] H. Doraiswamy and V. Natarajan. **Computing Reeb Graphs as a Union of Contour Trees**. *IEEE Transactions on Visualization and Computer Graphics*, **19**(2):249–262, 2013. 16

[38] Harish Doraiswamy, Nithin Shivashankar, Vijay Natarajan, and Yusu Wang. **Topological saliency**. *Computers & Graphics*, **37**(7):787–799, 2013. 17, 37

[39] Ian I. Dryden and Kantilal V. Mardia. *Statistical shape analysis*. Wiley, New York, 1998. 6, 7

[40] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2000. 52

[41] Herbert Edelsbrunner and Ernst Peter Mücke. **Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms**. *ACM Trans. Graph.*, **9**(1):66–104, 1990. 36

[42] R. Fabbri, L. D. F. Costa, J. C. Torelli, and O. M. Bruno. **2D Euclidean distance transform algorithms: A comparative survey**. *ACM Computing Surveys*, **40**(1):2:1–2:44, 2008. 39

[43] M. Fiedler. **Algebraic connectivity of graphs**. *Czechoslovak Mathematical Journal*, **23**(98):298–305, 1973. 41

[44] A.J. Fields and T.M. Keaveny. **Trabecular architecture and vertebral fragility in osteoporosis**. *Current Osteoporosis Reports*, **10**(2):132–140, 2012. 55

[45] H. Freeman and S.P. Morse. **On searching a contour map for a given terrain elevation profile**. *Journal of the Franklin Institute*, **284**(1):1–25, July 1967. 16

[46] D. J. Hand and K. Yu. **Idiot's Bayes-Not So Stupid After All?** *International Statistical Review*, **69**(3):385–398, 2001. 52

[47] T. P. Harrigan and R. W. Mann. **Characterization of microstructural anisotropy in orthotropic materials using a second rank tensor**. *Journal of Materials Science*, **19**(3):761–767, 1984. 8

[48] William Harvey and Yusu Wang. **Topological Landscape Ensembles for Visualization of Scalar-Valued Functions**. *Computer Graphics Forum*, **29**(3):993–1002, 2010. 37

[49] Tobias Heimann and Hans-Peter Meinzer. **Statistical shape models for 3D medical image segmentation: A review**. *Medical Image Analysis*, **13**(4):543–563, August 2009. 6, 7

[50] T. Hildebrand and P. Ruegsegger. **Quantification of Bone Microarchitecture with the Structure Model Index**. *Computer Methods in Biomechanics and Biomedical Engineering*, **1**(1):15–23, 1997. 8

[51] Joseph M. Hovadik and David K. Larue. **Static characterizations of reservoirs: refining the concepts of connectivity and continuity**. *Petroleum Geoscience*, **13**(3):195–211, August 2007. 10, 11

[52] T. Itoh and K. Koyamada. **Automatic isosurface propagation using an extrema graph and sorted boundary cell lists**. *IEEE Transactions on Visualization and Computer Graphics*, **1**(4):319–327, 1995. 27

[53] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. **Three-dimensional shape searching: state-of-the-art review and future trends**. *Computer-Aided Design*, **37**(5):509–530, April 2005. 6

[54] Z. Jiang, W. Chen, and C. Burkhart. **Efficient 3D porous microstructure reconstruction via Gaussian random field and hybrid optimization**. *Journal of Microscopy*, **252**(2):135148, 2013. 54

[55] Anthony C. Jones, Christoph H. Arns, Dietmar W. Hutmacher, Bruce K. Milthorpe, Adrian P. Sheppard, and Mark A. Knackstedt. **The correlation of pore morphology, interconnectivity and physical properties of 3D ceramic scaffolds with bone ingrowth**. *Biomaterials*, **30**(7):1440–1451, March 2009. 1

[56] Lutz Kettner, Jarek Rossignac, and Jack Snoeyink. **The safari interface for visualizing time-dependent volume data using iso-surfaces and contour spectra**. *Comput. Geom. Theory Appl.*, **25**(1-2):97–116, 2003. 16

[57] C. Knudby and J. Carrera. **On the relationship between indicators of geostatistical, flow and transport connectivity**. *Advances in Water Resources*, **28**(4):405–421, 2005. 9

[58] I.S. Kweon and T. Kanade. **Extracting Topographic Terrain Features from Elevation Maps**. *CVGIP: Image Understanding*, **59**(2):171–182, 1994. 16

[59] D. Legland, K. Kiêu, and M. F. Devaux. **Computation of Minkowski measures on 2D and 3D binary images**. *Image Analysis & Stereology*, **26**(2):83–92, 2007. 10

[60] C. L. Lin and J. D. Miller. **Network analysis of filter cake pore structure by high resolution X-ray microtomography**. *Chemical Engineering Journal*, **77**(1–2):79–86, 2000. 9

[61] C. L. Lin and J. D. Miller. **Pore structure analysis of particle beds for fluid transport simulation during filtration**. *International Journal of Mineral Processing*, **73**(2–4):281–294, 2004. 9

[62] P. J. Linstrom and W. G. Mallard. **NIST Chemistry WebBook, NIST Standard Reference Database Number 69**. http://webbook.nist.gov, National Institute of Standards and Technology, Gaithersburg MD, 20899, retrieved October 10, 2013. 64

[63] X. Liu and J.R. Ramirez. **Automated vectorization and labeling of very large raster hypsographic map images using a contour graph**. *Surveying and Land Information Systems*, **57**(1):5–10, 1997. 16

[64] Sven Loncaric. **A survey of shape analysis techniques**. *Pattern Recognition*, **31**(8):983–1001, August 1998. 6

[65] N. M. Maia de Abreu. **Old and new results on algebraic connectivity of graphs**. *Linear Algebra and its Applications*, **423**(1):53–73, 2007. 44, 75

[66] T. J. Mäkinen, J. J. Alm, H. Laine, E. Svedström, and H. T. Aro. **The incidence of osteopenia and osteoporosis in women with hip osteoarthritis scheduled for cementless total joint replacement**. *Bone*, **40**(4):1041–1047, 2007. 57

[67] A. A. Malcolm, H. Y. Leong, A. C. Spowage, and A. P. Shacklock. **Image segmentation and analysis for porosity measurement**. *Journal of Materials Processing Technology*, **192–193**:391–396, 2007. 9

[68] P. Maragos. **Pattern spectrum and multiscale shape representation**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**(7):701–716, 1989. 8

[69] C. R. J. Maurer, R. Qi, and V. Raghavan. **A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **25**(2):265 – 270, 2003. 39

[70] Joseph Mecke and Dietrich Stoyan. **The specific connectivity number of random networks**. *Advances in Applied Probability*, **33**(3):576–585, September 2001. 11

[71] John Willard Milnor. *Morse Theory*. Princeton University Press, 1963. 18, 22

[72] S. Mizuta and T. Matsuda. **Description of Digital Images by Region-Based Contour Trees**. In *Image Analysis and Recognition*, **3656** of *Lecture Notes in Computer Science*, pages 549–558. Springer Berlin / Heidelberg, 2005. 27

[73] J. J. Molitierno. *Applications of Combinatorial Matrix Theory to Laplacian Matrices of Graphs*. Taylor & Francis, 2012. 44, 75

[74] P. Monasse and F. Guichard. **Fast Computation of a Contrast-Invariant Image Representation**. *IEEE Trans. on Image Proc*, **9**:860–872, 1998. 5, 73

[75] Michael J. Moore, Esmaiel Jabbari, Erik L. Ritman, Lichun Lu, Bradford L. Currier, Anthony J. Windebank, and Michael J. Yaszemski. **Quantitative analysis of interconnectivity of porous biodegradable scaffolds with micro-computed tomography**. *Journal of Biomedical Materials Research Part A*, **71A**(2):258–267, 2004. 13

[76] N. Moritz, J. J. Alm, P. Lankinen, T. J. Mäkinen, K. Mattila, and H. T. Aro. **Quality of intertrochanteric cancellous bone as predictor of femoral stem RSA migration in cementless total hip arthroplasty**. *Journal of Biomechanics*, **44**(2):221–227, 2011. 57, 59

[77] S. Neethirajan, D. S. Jayas, N. D. G. White, and H. Zhang. **Investigation of 3D geometry of bulk wheat and pea pores using X-ray computed tomography images**. *Computers and Electronics in Agriculture*, **63**(2):104–111, 2008. 9

[78] S. Neethirajan, C. Karunakaran, D. S. Jayas, and N. D. G. White. **X-ray Computed Tomography Image Analysis to explain the Airflow Resistance Differences in Grain Bulks**. *Biosystems Engineering*, **94**(4):545–555, 2006. 9

[79] A. Odgaard. **Three-dimensional methods for quantification of cancellous bone architecture**. *Bone*, **20**(4):315–328, April 1997. 8

[80] A. Odgaard and H. J. G. Gundersen. **Quantification of connectivity in cancellous bone, with special emphasis on 3-D reconstructions**. *Bone*, **14**(2):173–182, 1993. 10, 55

[81] P. Oesterling, C. Heine, H. Janicke, and G. Scheuermann. **Visual analysis of high dimensional point clouds using topological landscapes**. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 113–120, 2010. 37

[82] J. Ohser and K. Schladitz. *3D Images of Materials Structures: Processing and Analysis*. Wiley-VCH, 2009. 8, 11, 51

[83] Norifumi OKII, Shigeru NISHIMURA, Kaoru KURISU, Yukio TAKESHIMA, and Tohru UOZUMI. **In Vivo Histological Changes Occurring in Hydroxyapatite Cranial Reconstruction**. *Neurologia medico-chirurgica*, **41**(2):100–104, 2001. 11

[84] D. W. Paglieroni. **Distance transforms: Properties and machine vision applications**. *CVGIP: Graphical Models and Image Processing*, **54**(1):56–74, 1992. 38

[85] V. Pascucci and K. Cole-McLaughlin. **Efficient computation of the topology of level sets**. In *IEEE Visualization, 2002. VIS 2002*, pages 187 –194, 2002. 16, 28

[86] Valerio Pascucci and Kree Cole-McLaughlin. **Parallel Computation of the Topology of Level Sets**. *Algorithmica*, **38**(1):249–268, 2004. 28

[87] Valerio Pascucci, Kree Cole-McLaughlin, and Giorgio Scorzelli. **The TOPORRERY: computation and presentation of multi-resolution topology**. In Torsten Möller, Bernd Hamann, and Robert D. Russell, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization, pages 19–40. Springer Berlin Heidelberg, 2009. 37

[88] S.V Patankar and D.B Spalding. **A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows**. *International Journal of Heat and Mass Transfer*, **15**(10):1787–1806, 1972. 64

[89] Jörg Petrasch, Fabian Meier, Hansmartin Friess, and Aldo Steinfeld. **Tomography based determination of permeability, Dupuit-Forchheimer coefficient, and interfacial heat transfer coefficient in reticulate porous ceramics**. *International Journal of Heat and Fluid Flow*, **29**(1):315–326, 2008. 63

[90] Sanjay Rana (ed.). *Topological data structures for surfaces: An Introduction to Geographical Information Science*. John Wiley & Sons, 2004. 73

[91] Georges Reeb. **Sur les points singuliers d'une forme de Pfaff compltement intgrable ou d'une fonction numrique**. *Comptes Rendus Acad. Sciences*, **222**:847–849, 1946. 7

[92] Philippe Renard and Denis Allard. **Connectivity metrics for subsurface flow and transport**. *Advances in Water Resources*, **51**:168–196, 2013. 9, 10, 12

[93] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. **Laplace-Beltrami spectra as Shape-DNA of surfaces and solids**. *Computer-Aided Design*, **38**(4):342–366, April 2006. 7

[94] A. Rosenfeld and J. L. Pfaltz. **Sequential Operations in Digital Picture Processing**. *Journal of the Association for Computing Machinery*, **13**(4):471–494, 1966. 38

[95] Azriel Rosenfeld. **Connectivity in Digital Pictures**. *J. ACM*, **17**(1):146160, January 1970. 5, 29

[96] Raif M. Rustamov. **Laplace-Beltrami Eigenfunctions for Deformation Invariant Shape Representation**. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, page 225233, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. 7

[97] P. Salembier, A. Oliveras, and L. Garrido. **Antiextensive connected operators for image and sequence processing**. *IEEE Transactions on Image Processing*, **7**(4):555 –570, 1998. 73

[98] P. Salembier and M. H. F. Wilkinson. **Connected operators**. *Signal Processing Magazine, IEEE*, **26**(6):136 –157, 2009. 5

[99] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. **A review of recent range image registration methods with accuracy evaluation**. *Image and Vision Computing*, **25**(5):578–596, May 2007. 6

[100] E. Santos, J. Tierny, A. Khan, B. Grimm, L. Lins, J. Freire, V. Pascucci, C.T. Silva, S. Klasky, R. Barreto, and N. Podhorszki. **Enabling Advanced Visualization Tools in a Web-Based Simulation Monitoring System**. In *Fifth IEEE International Conference on e-Science, 2009. e-Science '09*, pages 358–365, 2009. 17

[101] K. Schladitz. **Quantitative micro-CT**. *Journal of Microscopy*, **243**(2):111–117, 2011. 12

[102] D. Schneider, C. Heine, H. Carr, and G. Scheuermann. **Interactive comparison of multifield scalar data based on largest contours**. *Computer Aided Geometric Design*, 2012. (in press). 17

[103] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. **Interactive Comparison of Scalar Fields Based on Largest Contours with Applications to Flow Visualization**. *Visualization and Comp. Graphics, IEEE Trans. on*, **14**(6):1475–1482, 2008. 17

[104] C. Scholz, F. Wirner, J. Götz, U. Rüde, G.E. Schröder-Turk, K. Mecke, and C. Bechinger. **Permeability of porous materials determined from the Euler characteristic**. *Physical Review Letters*, **109**(26), 2012. 11

[105] T.B. Sebastian, P.N. Klein, and B.B. Kimia. **Recognition of shapes by editing their shock graphs**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**(5):550–571, 2004. 6

[106] A. P. Sheppard, R. M. Sok, and H. Averdunk. **Techniques for image enhancement and segmentation of tomographic images of porous materials**. *Physica A: Statistical Mechanics and its Applications*, **339**(1–2):145–151, 2004. 9

[107] J. Shi and J. Malik. **Normalized cuts and image segmentation**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(8):888 –905, 2000. 41

[108] J. Sklansky. **Measuring Concavity on a Rectangular Mosaic**. *IEEE Transactions on Computers*, **C-21**(12):1355–1364, December 1972. 73

[109] Stephen Smale. **On Gradient Dynamical Systems**. *Annals of Mathematics*, **74**(1):199–206, July 1961. 7

[110] B.-S. Sohn and C. Bajaj. **Time-varying contour topology**. *IEEE Transactions on Visualization and Computer Graphics*, **12**(1):14–25, 2006. 17

[111] Dietrich Stauffer and Ammon Aharony. *Introduction To Percolation Theory*. Taylor & Francis, July 1994. 11

[112] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. **A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion**. *Computer Graphics Forum*, **28**(5):1383–1392, 2009. 7

[113] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. **Skeleton based shape matching and retrieval**. In *Shape Modeling International*, pages 130–139, 2003. 7

[114] Shigeo Takahashi, Tetsuya Ikeda, Yoshihisa Shinagawa, Tosiyasu L. Kunii, and Minoru Ueda. **Algorithms for Extracting Correct Critical Points and Constructing Topological Graphs from Discrete Geographical Elevation Data**. *Computer Graphics Forum*, **14**(3):181–192, 1995. 16, 27

[115] Shigeo Takahashi, Yuriko Takeshima, and Issei Fujishiro. **Topological volume skeletonization and its application to transfer function design**. *Graph. Models*, **66**(1):24–49, January 2004. 16

[116] JohanW.H. Tangelder and RemcoC. Veltkamp. **A survey of content based 3D shape retrieval methods**. *Multimedia Tools and Applications*, **39**(3):441–471, 2008. 6

[117] Sergey P. Tarasov and Michael N. Vyalyi. **Construction of contour trees in 3D in O(nlogn) steps**. In *Proceedings of the fourteenth annual symposium on Computational geometry*, SCG '98, pages 68–75, New York, NY, USA, 1998. ACM. 27

[118] S. Tassani, C. Öhman, M. Baleani, F. Baruffaldi, and M. Viceconti. **Anisotropy and inhomogeneity of the trabecular structure can describe the mechanical strength of osteoarthritic cancellous bone**. *Journal of Biomechanics*, **43**(6):1160–1166, 2010. 61, 77

[119] D.M. Thomas and V. Natarajan. **Symmetry in Scalar Field Topology**. *IEEE Transactions on Visualization and Computer Graphics*, **17**(12):2035–2044, 2011. 17

[120] J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci. **Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees**. *IEEE Transactions on Visualization and Computer Graphics*, **15**(6):1177–1184, 2009. 16

[121] S. Torquato, J. D. Beasley, and Y. C. Chiew. **Twopoint cluster function for continuum percolation**. *The Journal of Chemical Physics*, **88**(10):6540–6547, May 1988. 10

[122] C. Tzafestas and P. Maragos. **Shape Connectivity: Multiscale Analysis and Application to Generalized Granulometries**. *Journal of Mathematical Imaging and Vision*, **17**:109–129, 2002. 8

[123] J. K. Udupa and S. Samarasekera. **Fuzzy Connectedness and Object Definition: Theory, Algorithms, and Applications in Image Segmentation**. *Graphical Models And Image Processing*, **58**(3):246–261, 1996. 6

[124] 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 Transactions on Pattern Analysis and Machine Intelligence*, **29**(2):272–285, 2007. 8

[125] Marc van Kreveld, René van Oostrum, Chandrajit Bajaj, Valerio Pascucci, and Dan Schikore. **Contour trees and small seed sets for isosurface traversal**. In *Proceedings of the thirteenth annual symposium on Computational geometry*, SCG '97, pages 212–220, New York, NY, USA, 1997. ACM. 26, 27

[126] R.C. Veltkamp. **Shape matching: similarity measures and algorithms**. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 188–197, 2001. 6

[127] L. Vincent. **Granulometries and opening trees**. *Fundamenta Informaticae*, **41**(1-2):57–90, 2000. 8

[128] H. J. Vogel. **Morphological determination of pore connectivity as a function of pore size using serial sections**. *European Journal of Soil Science*, **48**(3):365–377, 1997. 10, 11, 13

[129] H. J. Vogel, U. Weller, and S. Schlüter. **Quantification of soil structure based on Minkowski functions**. *Computers & Geosciences*, **36**(10):1236–1245, 2010. 11, 13

[130] G. H. Weber, S. E. Dillard, H. Carr, V. Pascucci, and B. Hamann. **Topology-controlled volume rendering**. *IEEE Transactions on Visualization and Computer Graphics*, **13**(2):330–341, 2007. 16

[131] G.H. Weber, P.-T. Bremer, and V. Pascucci. **Topological Landscapes: A Terrain Metaphor for Scientific Data**. *IEEE Transactions on Visualization and Computer Graphics*, **13**(6):1416–1423, 2007. 37

[132] Gunther H. Weber, Peer-Timo Bremer, and Valerio Pascucci. **Topological Cacti: Visualizing Contour-Based Statistics**. In Ronald Peikert, Helwig Hauser, Hamish Carr, and Raphael Fuchs, editors, *Topological Methods in Data Analysis and Visualization II*, Mathematics and Visualization, pages 63–76. Springer Berlin Heidelberg, 2012. 37

[133] M. H. F. Wilkinson. **Attribute-space connectivity and connected filters**. *Image and Vision Computing*, **25**(4):426–435, 2007. 5

[134] M. H. F. Wilkinson, G. K. Ouzounis, and P. W. Hawkes. **Chapter 5 - Advances in Connectivity and Connected Attribute Filters**. In *Advances in Imaging and Electron Physics*, **161**, pages 211–275. Elsevier, 2010. 5

[135] R. C. Wilson, E. R. Hancock, and B. Luo. **Pattern vectors from algebraic graph theory**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **27**(7):1112 –1124, 2005. 41

[136] W. C. Yueh. **Eigenvalues of Several Tridiagonal Matrices**. *Appl Math ENotes*, **5**:66–74, 2005. 44

[137] Dengsheng Zhang and Guojun Lu. **Review of shape representation and description techniques**. *Pattern Recognition*, **37**(1):1–19, January 2004. 6

[138] X. Zhang, C. L. Bajaj, and N. Baker. **Fast Matching of Volumetric Functions Using Multi-resolution Dual Contour Trees**. Technical report, Texas Institute for Computational and Applied Mathematics, Austin, Texas, 2004. 17

[139] Z.-M. Zhang, Z.-C. Li, L.-S. Jiang, S.-D. Jiang, and L.-Y. Dai. **Micro-CT and mechanical evaluation of subchondral trabecular bone structure between postmenopausal women with osteoarthritis and osteoporosis**. *Osteoporosis International*, **21**(8):1383–1390, August 2010. 1

[140] Jianlong Zhou and M. Takatsuka. **Automatic Transfer Function Generation Using Contour Tree Controlled Residue Flow Model and Color Harmonics**. *IEEE Transactions on Visualization and Computer Graphics*, **15**(6):1481–1488, 2009. 16

[141] Jianlong Zhou and M. Takatsuka. **Importance Driven Contour Tree Simplification**. In *2011 International Conference on Internet Computing Information Services (ICICIS)*, pages 265–268, 2011. 36, 37

[142] Jianlong Zhou and Chun Xiao. **Similarity-driven multi-level partial contour tree simplification**. In *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 1949–1953, 2012. 36

[143] Afra Zomorodian. **Algorithms and Theory of Computation Handbook**. page 33. Chapman & Hall/CRC, 2010. 7, 22

ORIGINAL PUBLICATIONS

PUBLICATION-I

Aydogan D.B. & Hyttinen J.

Binary image representation by contour trees

Proceedings of SPIE 8314, Medical Imaging 2012: Image Processing, 83142X,
4-10.02.2012, San Diego, CA, USA

Binary image representation by contour trees

Dogu Baran Aydogan*[a] and Prof. Jari Hyttinen[a]

[a]Dept. of Biomedical Eng./Tampere Uni. of Tech., Finn-Medi 1 L 4, Biokatu 6, 33520 Tampere, Finland

ABSTRACT

There is a growing need in medical image processing to analyze segmented objects. In this study we are interested in analyzing morphological properties of complex structures such as the trabecular bone. Although, there are various shape description approaches proposed in the literature, there is not an adequate method to represent foreground object(s) morphology with respect to the background.

In this article, we propose a way of representing binary images of any dimensions using graphs that emphasize connectivity of level-sets to foreground and background. We start by calculating the euclidean distance transform (EDT) to create a scalar field. Then the contour tree of this scalar field is calculated using a modified version of the algorithm proposed by Carr[1]. Contour trees are mostly used to visualize high dimensional scalar fields as they can put on view the critical points, i.e: local min, max and saddle points; however, their use in representing complex shapes have not been studied. We demonstrate the use of our method on artificial 2D images having different topologies as well as 3D μ-CT images of two bone biopsies.

We show that the application of contour trees to complex binary data particularly prove useful when interpreting pore-networks at micro-scale. Further work to quantify foreground and background interconnectivity using certain graph theoretical methods is still under research.

Keywords: binary image representation, contour trees

1. INTRODUCTION

In many fields of medical image processing, there is a growing need of analyzing foreground objects with respect to background. For example, after segmentation, 3D microcomputed-tomography (μ-CT) images of foam-like materials such as trabecular bones reveal interconnected networks of pore spaces as well as the bone itself. Similarly, confocal microscopy images of biological tissues put on view individual cells that compose them. Generally, the information that is seeked is two-fold (i) the topological properties of the foreground object(s) and (ii) how foreground objects are placed on the background.

There are several shape descriptors in the literature to quantify morphological properties of segmented objects. A review of these methods can be found in [2]. However almost all of these approaches work for a single body. On the other hand, multiple objects are conventionally analyzed by connected component labeling algorithms which are followed by analyzing each label seperately by the aforementioned shape descriptors. Although there has been over a decade long history in the study of binary data, structure related parameters that are currently used to characterize complex tissues or biomaterials are still mostly qualitative, vaguely defined and not suitable for accurately describing connectivity information. In this work, we propose a new and balanced approach to visualize and track the morphological properties of foreground objects with regard to their placement on the background.

Previous research in this field include [3], in which authors use a watershedding based approach for quantifying interconnectivity for foam materials' porous space. In [4], for the same goal, a more heuristic approach is proposed based on iteratively labeling connected components and applying morphological opening operator. [5] proposes using the medial axis to partition pore-space and calculate strut thicknesses of foam materials. Medial axis is also used in [6] for characterizing foreground fiber network topology. On the other hand, pore-space formed within fibrous materials has not been studied as much. An excellent collection of the previous research done regarding micro-structure analysis of materials and the pore space within is gathered in [7].

E-mail: baran.aydogan@tut.fi, Telephone: +358 40 849 0163

Medical Imaging 2012: Image Processing, edited by David R. Haynor, Sébastien Ourselin,
Proc. of SPIE Vol. 8314, 83142X · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.911493

Although there has been over a decade long history in the study of such data, structure related parameters that are currently used to characterize complex tissues or biomaterials are still mostly qualitative, vaguely defined and not suitable for accurately describing connectivity information. The quantitative approaches that are briefly summarized in the previous paragraph only work either for the background or the foreground object(s). Moreover, in order to compare morphological properties of different types of geometries, such as in foam and fibrious materials, there is a need to develop a single approach that works for all. To our knowledge, there is not an adequate approach in the literature that is capable of working for any geometry in describing foreground and background morphology jointly. The aim of this paper is to fill this gap by introducing a new way of describing shapes via contour trees which is suitable for any dimension of data and any topology.

2. METHODS AND MATERIALS

2.1 Overview of the approach

In this work, we propose to represent binary images by contour trees after computing the Euclidean distance transform (EDT).

Let $\mathbf{X}$ be the domain of a binary image $V(X)$ and let $x, y \in \mathbf{X}$. Then $\forall x \in \mathbf{X}$, EDT maps to the distance given by $F(x) = min\{\parallel x - y \parallel : V(y) = 0 \wedge 1\}$. The final scalar field to be used with the contour tree algorithm was obtained by assigning positive distance values for the foreground object(s) and negative values for the background.

Therefore, the contour tree shows the global and local maximum values for foreground objects(s) whereas global and local minimum for the background. As a result the complete tree represents the connectivity of foreground objects with regard to their placement on the background.

2.2 Contour trees

Isocontours have been used for over a century to visualize scalar fields, such as for topographic analysis in [8]. For higher dimensional data, isosurfaces have been used intensively during the last decades for the same purpose. The main aim of contour trees is to follow the evolution of isocontours(/isosurfaces) as they appear, join, split or dissappear by means of a graph. Contour trees are a special type of Reeb graphs and used mainly for the visualization of 3D scalar fields [9].

In this study, we adopted the contour tree calculation algorithm given in [1]. The algorithm starts by computing the "join and split trees"; in a later stage of the algorithm these are merged to form the contour tree.

Let $x \in \mathbf{X}$ be the domain of the scalar field $F(\mathbf{X})$. Let l represent any level between the global-min ($l_{min} \in F(\mathbf{X})$) and global-max ($l_{max} \in F(\mathbf{X})$).

For the join tree calculation, l is swept from l_{max} towards l_{min} to represent the connectivity of regions $x : F(\mathbf{X}) \leq l$ in the form of a tree graph. For the split tree calculation, l is swept from l_{min} towards l_{max} to represent the connectivity of regions $x : F(\mathbf{X}) \geq l$.

Figure-1 shows the flowchart of procedures in order to obtain the contour tree from a binary input image. Image is first transform to a scalar field using EDT. The split and join trees are calculated using at the level set values given in "Levels" input. Finally, two trees are merged to form the contour tree.

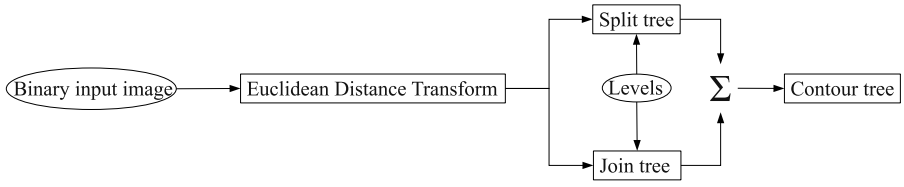


Figure 1. The procedure to obtain the contour tree from a binary input image

In the original article [1], the values in $F(\mathbf{X})$ are sorted and $l = F(x)$ are used when obtaining the join and split trees. However, this approach creates numerous unwanted edges in the contour tree as a result of the inclusion of all the critical points. The authors of the algorithm also mention this problem in their work and propose several approaches to remove this noise in the contour tree.

In our work, in order to avoid the noise in the contour tree, we quantized the values of l between l_{min} and l_{max} instead of tracing all the vertices in the image. The quantization step size is determined by the user. One other reason for supplying the levels seperately was that, number of vertices decrease significantly in this manner as compared to the number of vertices in the image. Therefore, it makes it easier and faster to process the augmented contour tree later on.

2.3 Test images

We tested our method on both artificial images and real data. We used the 2D images to demostrate how the topological differences are reflected on the contour tree. In addition to the artificial images, we used μ-CT images of normal and cancellous trabecular bone samples. All the images that are used in the study are shown in Figure-2.

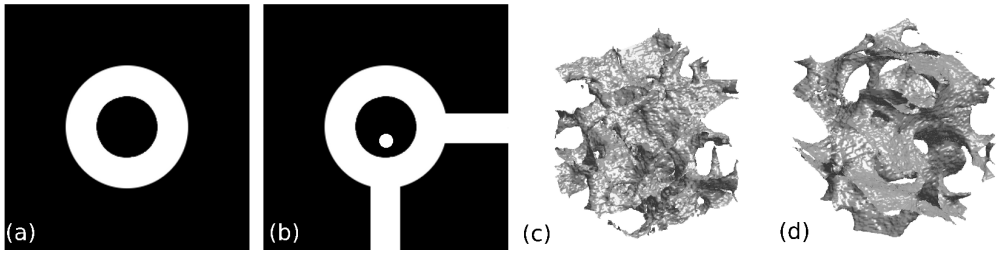


Figure 2. Test images that are used for the study. 2D artificial images (a,b) and 3D trabecular bone samples osteoporotic (c) and healthy (d)

3. RESULTS

Figure-3 shows the calculated EDT image for the artificial 2D sample shown in Figure-2 (b). For a better visualization of how split and join trees are calculated from this data, the EDT image is also shown in 3D by putting the intensity values as levels of elevation in Figure-3.

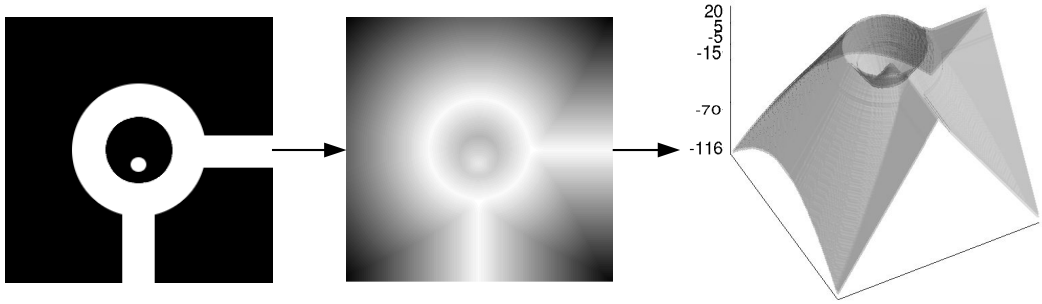


Figure 3. EDT image (middle) obtained from the 2D artificial sample (left). EDT image is also shown as a 3D elevation map (right).

Figure-4 shows how the split and join trees are calculated from the sample shown in Figure-2 (b) on the 3D elevation map. First two coloumns of Figure-4 show the formation of split tree as the level-sets join from global minimum to global maximum. Through the evolution of the split tree local minimums are caught. The two coloumns on the right of Figure-4 show how the join tree evolves from the global maximum to the global minimum. Local maximum at level 5 is caught by the join tree.

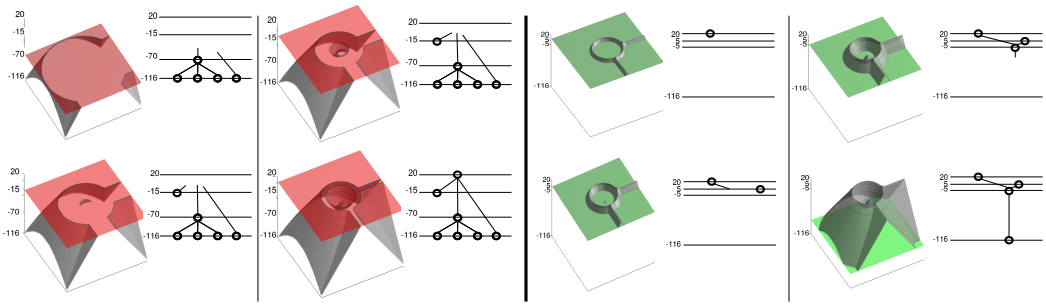


Figure 4. Split and join trees used in extracting the contour tree. Left hand side shows how the split tree forms; right hand side shows how the join tree forms.

The final contour tree representing the 2D artificial sample in Figure-2 (b) is calculated by merging the split and the join trees according to the algorithm described in [1]. For this data, contour tree is calculated using levels varying from global maximum (20) to global minimum (-116) with a step size of 1. Figure-5 shows the corresponding contour tree. (The augmentation step is skipped in Figure-5)

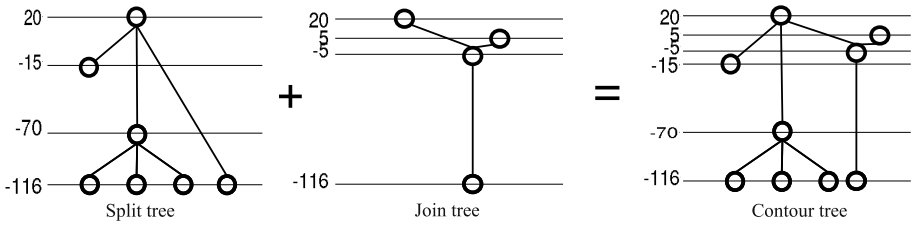


Figure 5. Contour tree (right) as a fusion of split (left) and join (middle) trees

In Fig-6 the contour tree is calculated for the image in Figure-2 (a). The isolevels are varied by a step size of 1 for this data as well.

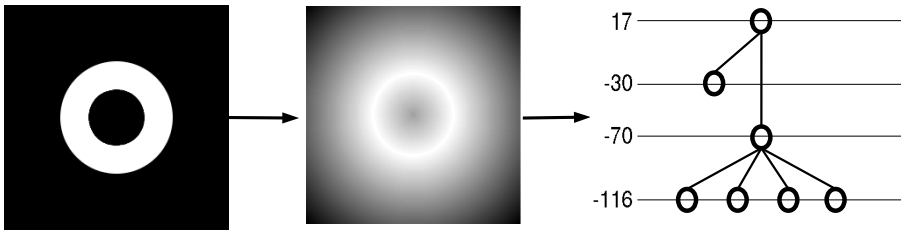


Figure 6. EDT image (middle) and corresponing contour tree (right) for the test sample given in Fig-2 (a)

For the bone samples, contour trees are drawn in Fig-6. For both of the bone data, contour trees are calculated using isolevels varying from 8 to -32 with a step size of 8.

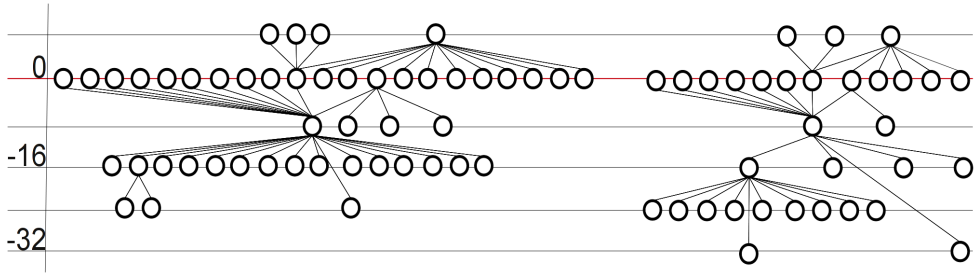


Figure 7. Results for the bone samples. Osteoporotic bone contour tree (left) and healthy bone contour (right)

4. DISCUSSIONS AND CONCLUSIONS

In this work, we proposed a new approach to represent binary images using contours trees. We used artificial 2D data for the demonstration and 3D μ-CT images of bone biopsies as real case studies. We adopted a fast contour tree computation algorithm and modified it for our needs in order to avoid extra edges in the graph.

The calculated contour trees shown in Figures - 5 and 6 put on view the differences between the two artificial structures. In the case of a single connected component (Figure-2 (a)), the graph represents how thick regions of the foreground object are getting connected/split with the rest of the foreground object as well as the the background. This representation is also able to show how big is the background and how disconnected it becomes with the foreground object. In the case of multiple connected components (Figure-2 (b)), the graph also shows how much disconnected objects are seperated. With the work on artificial data, contour tree representations of binary images were shown to be useful tools in understanding topological properties of structures.

The results shown for μ-CT images of normal and cancellous trabecular bone samples in Figure-7 clearly show differences in the contour tree. This result was expected as cancellous bone has more disconnected components in its microstructure.

Binning l values between l_{min} and l_{max} succesfully eliminates large numbers of edges in the contour tree but this comes with a cost. Saddle points are not calculated accurately. However, for our purpose, the interest is not the locations of the saddle points; the information we are after is whether a saddle occurs between two levels or not. Therefore, the cost of mislocating saddle points is worth avoiding the contour tree pruning.

As the conclusion, the initial results that are given in this paper are promising. We are continuing this research towards quantifying an interconnectivity parameter using the calculated contour trees. The results of the ongoing research for interconnectivity quantification will be used in the diagnosis of osteoporosis and other similar diseases which show themselves as structural deformations in tissues.

ACKNOWLEDGMENTS

This worked is funded by the TEKES (Finnish Funding Agency for Technology and Innovation (www.tekes.fi)) project MIKAMA II and Jenny & Antti Wihuri Foundation. The authors thank Niko Moritz (University of Turku, Finland) for μ-CT image acquisition and reconstruction.

REFERENCES

[1] Carr, H., Snoeyink, J., and Axen, U., "Computing contour trees in all dimensions," in [*Proceedings of the eleventh annual symposium on Discrete algorithms*], 918–926, Society for Industrial and Applied Mathematics, San Francisco, California, United States (2000).

[2] D., Z., "Review of shape representation and description techniques," *Pattern Recognition* **37**(1), 1–19 (2004).

[3] Atwood, R., Jones, J., Lee, P., and Hench, L., "Analysis of pore interconnectivity in bioactive glass foams using x-ray microtomography," *Scripta Materialia* **51**(11), 1029–1033 (2004).

[4] Moore, M. J., Jabbari, E., Ritman, E. L., Lu, L., Currier, B. L., Windebank, A. J., and Yaszemski, M. J., "Quantitative analysis of interconnectivity of porous biodegradable scaffolds with micro-computed tomography," *Journal of Biomedical Materials Research Part A* **71A**(2), 258–267 (2004).

[5] Jones, A. C., Arns, C. H., Hutmacher, D. W., Milthorpe, B. K., Sheppard, A. P., and Knackstedt, M. A., "The correlation of pore morphology, interconnectivity and physical properties of 3D ceramic scaffolds with bone ingrowth," *Biomaterials* **30**, 1440–1451 (Mar. 2009).

[6] D'Amore, A., Stella, J. A., Wagner, W. R., and Sacks, M. S., "Characterization of the complete fiber network topology of planar fibrous tissues and scaffolds," *Biomaterials* **31**, 5345–5354 (July 2010).

[7] Ohser, J. and Schladitz, K., [*3D Images of Materials Structures: Processing and Analysis*], Wiley-VCH (2009).

[8] Cayley, A., "On contour and slope lines (1859)," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **18**, 264–268.

[9] Carr, H. and Snoeyink, I., "Path seeds and flexible isosurfaces: Using topology for exploratory visualization," *Proceedings of Eurographics Visualization Symposium* , 49–58 (2003).

PUBLICATION-II

Aydogan D.B. & Hyttinen J.

Contour tree connectivity of binary images from algebraic graph theory

IEEE International Conference on Image Processing (ICIP),

15-22.09.2013, Melbourne, Australia

CONTOUR TREE CONNECTIVITY OF BINARY IMAGES FROM ALGEBRAIC GRAPH THEORY

Dogu Baran Aydogan and Jari Hyttinen

Tampere University of Technology
Department of Electronics and Communications Engineering
and BioMediTech, Tampere

ABSTRACT

We propose a novel feature for binary images that provides connectivity information by taking into account the proximity of connected components and cavities. We start by applying the Euclidean distance transform and then we compute the contour tree. Finally, we assign the normalized algebraic connectivity of a contour tree derivative as a feature for connectivity. Our algorithm can be applied to any dimensions of data as well as topology. And the resultant connectivity index is a single real number between 0 and 1.

We test and demonstrate interesting properties of our approach on various 2D and 3D images. With its intriguing properties, the proposed index is widely applicable for studying binary morphology. Especially, it is complementary to Euler number for studying connectivity of microstructures of materials such as soil, paper, filter, food products as well as biomaterials and biological tissues.

Index Terms— connectivity, contour tree, feature extraction, algebraic graph theory, binary morphology

1. INTRODUCTION

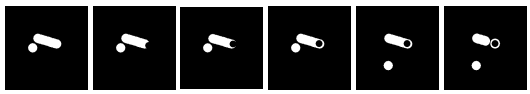
Connectivity has long been recognized as an important tool for analyzing images. The most commonly used connectivity measure for binary images is the Euler number (χ), which is expressed with the Euler-Poincaré formula as the alternating sum of Betti numbers, β , $\chi = \sum_k (-1)^k \beta_k$. For 2D, this is simply the number of connected components (β_0), minus the number of holes (β_1). For 3D, it is the number of connected components (β_0), minus the number of tunnels (β_1), plus the number of voids (β_2). Euler number is very easy to understand which makes it an elegant topological invariant. On the other hand, for most practical purposes, topology alone is not sufficient to describe connectivity properties of binary images. For example, although Euler number includes information regarding the number of connected components, it does not provide information about their proximity to each other. Similarly, Euler number can detect holes but it ignores the information regarding cavities.

This work is funded by the Jenny & Antti Wihuri Foundation.

Different aspects of connectivity have been investigated due to the increase in the diversity of applications and image processing problems. For example, based on connected operators [1, 2] and filters, hierarchical representations of regions are generated in [3, 4]. In order to represent the inclusion relationship of shapes, tree of shapes has been proposed in [5]. Another interpretation of connectivity is given in [6], where images are considered as fuzzy in nature. For segmentation applications, hanging togetherness is emphasized, which leads to the definition of fuzzy connected components.

In our earlier work, we proposed an approach to represent binary images using contour trees [7]. The main contribution of this work is a novel connectivity index that is derived from the contour tree representation of binary images. Our work is also the first in which spectral graph theory is used with contour trees.

In this paper, we introduce the contour tree connectivity (CTC), which provides connectivity information by taking into account the proximity of connected components and cavities. In order to provide an easy to use figure, we designed the CTC so that it is a real number between 0 and 1. For example, Fig.1 shows that there is an 11% connectivity decrease in terms of CTC from left to right. The decrease occurs due to cavity formation (a-c), hole formation (d), separation of connected components (e) and increase in the number of connected components (f). In contrast, Euler number only varies when a topology change occurs in (d) and (f).



CTC=0.71 χ=2 (a)	CTC=0.70 χ=2 (b)	CTC=0.68 χ=2 (c)	CTC=0.64 χ=1 (d)	CTC=0.62 χ=1 (e)	CTC=0.60 χ=2 (f)

Fig. 1: The connectivity information provided by CTC shows a total of 11% connectivity decrease from left to right. CTC is decreasing due to cavity formation (a-c), hole formation (d), separation of connected components (e) and increase in the number of connected components (f). Notice that Euler number varies only in (d) and (f) where the topology changes.

2. CONTOUR TREE CONNECTIVITY (CTC)

2.1. Overview of the CTC feature extraction method

Fig.2 shows the flowchart for extracting the connectivity feature proposed in this article. Starting with a binary image, we create a real valued image using the Euclidean distance transform (EDT). This image is used as the input for calculating the join and split trees. These two trees are then merged to form the contour tree and the supplemented contour tree (SCT). Lastly, the algebraic connectivity of the SCT is calculated and normalized with a function of the range of the EDT image, which gives the CTC.

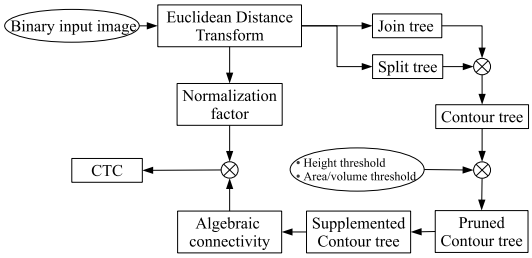


Fig. 2: Flowchart for the CTC feature extraction. From a binary input image, firstly EDT, then the SCT and finally the CTC is calculated as the normalized algebraic connectivity of SCT.

In subsections 2.2-2.7, the blocks in Fig.2 are explained.

2.2. Euclidean distance transform (EDT)

Euclidean distance transform converts binary images to gray scale in which the value of pixels are determined by their distance to the nearest non-object pixel. We used the signed Euclidean distance transform definition given in [8]. For the computation of EDT, we used Maurer's algorithm in [9].

2.3. Binary image representation using contour trees

The contour tree is a topological abstraction that follows the evolution of level sets as they appear, join, split or disappear by means of a graph. In [7], we proposed a representation for binary images which takes into account both the geometrical properties of foreground object(s) and how the object(s) is/are placed on the background. This is done by applying the Euclidean distance transform and by estimating its contour tree. Fig.3 demonstrates this representation on a sample image.

We implemented a sweep and merge method to compute the contour tree. Here, we initially compute the join and split trees and then merge them in a later step. Different than [7], in our current work, join/split tree computation and the merging are done by the approach given in [10, 11]. A contour tree that has only the critical points of an image (*i.e*: where topological changes occur between level sets) is also known as an *unaugmented* contour tree. An *augmented* version can be obtained by introducing vertices from other significant levels into the graph.

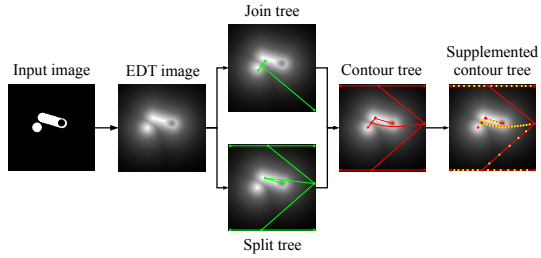


Fig. 3: Binary image representation using contour trees. First EDT, then join and split trees (green trees) are calculated. Join and split trees are merged to form the contour tree (red tree). SCT is shown on the right (red tree with yellow supplemented nodes).

2.4. Pruning

In order to remove the noise in the contour trees, we implemented a pruning algorithm that is based on three graph operations. These are (i) vertex collapse, (ii) leaf pruning and (iii) saddle collapse. The operations are shown in Fig. 4.

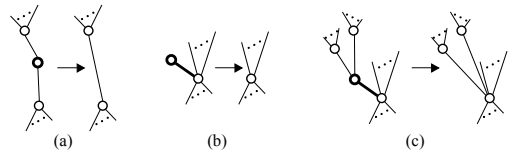


Fig. 4: Graph operations used for pruning contour trees (a) vertex collapse (b) leaf pruning (c) saddle collapse.

Vertex collapse removes only vertices and is done whenever possible. In order to perform leaf pruning or saddle collapse, the edge to be removed must have less area/volume than the defined *area/volume threshold* and the vertices on each sides of the edge should have a level difference below the given *level threshold*. The algorithm recursively removes unnecessary vertices and edges using these operations until no more removal is possible.

2.5. Supplemented contour tree (SCT)

In order to utilize our graph connectivity approach we need a particular tree that is similar to the augmented contour tree but has vertices only at integer valued levels. This tree is named the supplemented contour tree and is obtained as follows:

Definition 1. *The supplemented contour tree (SCT) of an image I is the graph, $\mathcal{G}(V, E)$, with vertex set V and edge set E that is obtained as follows:*

1. *Compute the contour tree, $\mathcal{T}(V^{\mathcal{T}}, E^{\mathcal{T}})$, of I.*
2. *For each $v_i^{\mathcal{T}} \in V^{\mathcal{T}}$ with level $\ell(v_i^{\mathcal{T}})$, create a new vertex $v_i' \in V'$ with a supplemented level $\lfloor \ell(v_i^{\mathcal{T}}) \rceil$ where $\lfloor . \rceil$ is the rounding operator.*
3. *Create an intermediate tree, $\mathcal{G}'(V', E')$ with $E' = E^{\mathcal{T}}$.*
4. *For each $e_i' \in E'$, initialize a new vertex set $V''_{e_i'} = \{\}$ and for each integer valued level along e_i', add a new vertex in $V''_{e_i'}$.*

5. *Create a path for each $e'_i \in E'$ to form an edge set $E''_{e'_i}$ by joining the vertices in $V''_{e'_i}$.*
6. *$\mathcal{G}(V, E)$ is the SCT with $V = \bigcup V''$ and $E = \bigcup E''$.*

SCT is a suitable graph representation for binary images as it is capable of showing how foreground/background regions are positioned in the image by the number of vertices along edges. The example in Fig. 3 shows the difference between the contour tree and the SCT.

2.6. Graph spectrum and algebraic connectivity

For a graph $\mathcal{G}$, the eigenvalues of the Laplacian matrix given as $L(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G})$, form the spectrum of $\mathcal{G}$. Here $D(\mathcal{G})$ and $A(\mathcal{G})$ are the degree diagonal and the adjacency matrices respectively. Note that for a connected graph, the eigenvalues of $L(\mathcal{G})$ can be sorted as $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. Since $\lambda_1 = 0$ for disconnected graphs, Fiedler gave the following definition for algebraic connectivity in [12] as a measure of graph connectivity:

Definition 2. *Algebraic connectivity of a graph $\mathcal{G}$, $\lambda_1(\mathcal{G})$, is the second smallest eigenvalue of its Laplacian matrix.*

2.7. Normalization - Upper and lower bounds

We are interested in the λ_1 bounds as a function of the height of the SCT. This makes it possible to compare λ_1 values using the following lower and upper bounds. From subsection 2.6 and Definition 2, we give the following lower bound:

Corollary 1. *$\mathcal{G}$ being an SCT, $\lambda_1(\mathcal{G}) > 0$ is a lower bound.*

The upper bound for λ_1 is given by the following theorem (proof is given in Appendix A):

Theorem 1. *$\mathcal{G}$ being an SCT, $\lambda_1(\mathcal{G}) \leq 2(1 - \cos\frac{\pi}{h+1})$ is an upper bound, where h is the height of $\mathcal{G}$.*

Finally, by using Definition 2, Corollary 1 and Theorem 1 we give the formal definition of the CTC as follows:

Definition 3. *Contour tree connectivity (CTC) of a binary image I is:*

$$\overline{\lambda_1(\mathcal{G})} = \frac{\lambda_1(\mathcal{G})}{2\left[1 - \cos\left(\frac{\pi}{\lfloor l^g_{max}\rceil - \lfloor l^g_{min}\rceil + 1}\right)\right]} \quad (1)$$

Here $\mathcal{G}$ is the SCT of I. λ_1 is the algebraic connectivity of $\mathcal{G}$; l^g_{max} and l^g_{min} are the global max and min of EDT of I.

In the rest of the text, we will use the abbreviation CTC and the symbol $\overline{\lambda_1}$ interchangeably. Lastly, we give the following corollary and conclude this section.

Corollary 2. *For any binary image, $0 < \overline{\lambda_1} \leq 1$.*

3. RESULTS AND DISCUSSIONS

3.1. 2D test images

We tested CTC with the synthetic dataset shown in Fig.5. The dataset is composed of 6 series (I-VI). Each series contains 6 images (A-F), that have slight changes in connectivity. All images are of dimensions 256×256. We prepared different cases for each series to show how $\overline{\lambda_1}$ varies. For example, series II shows the changes in $\overline{\lambda_1}$ during clumping of objects, III shows the changes during branching, IV shows the effect of holes.

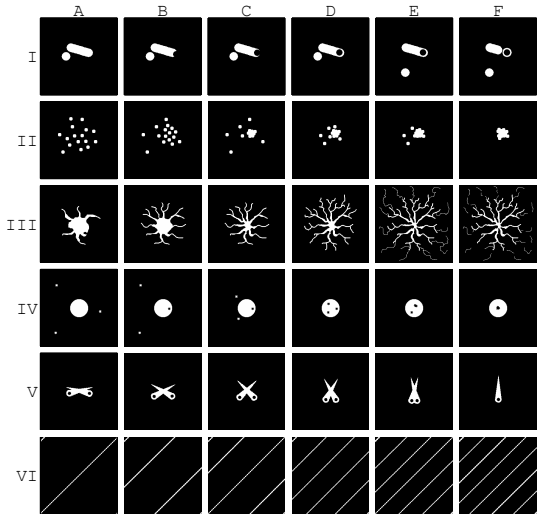


Fig. 5: 2D images used in the experiments. Each row contains a series of images. In each column connectivity is slightly altered.

We also tested CTC using the same images under various boundary noise conditions. To add boundary noise, we randomly moved the boundary pixels. We used Gaussian noise with $\mu = 0$ and $\sigma = \{1, 2, 3, 4\}$. Samples from the noisy images are shown in Fig.6. This type of noise allows for topological changes that may significantly alter Euler number.

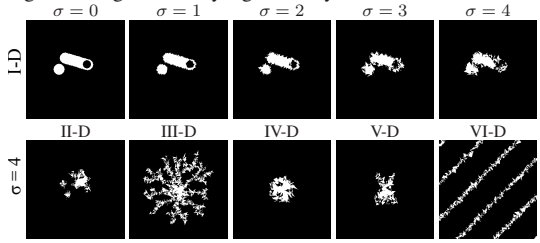


Fig. 6: Top row shows various levels of noise on image I-D. Bottom row shows the maximum noise for images II-D to VI-D.

Fig.7 shows the $\overline{\lambda_1}$ values for the test images. Error bars show the max and min values of $\overline{\lambda_1}$ for noisy images. It is observed from Fig.7 that $\overline{\lambda_1}$ consistently increases when foreground objects approach each other. This trend is best seen for series II. Also $\overline{\lambda_1}$ changes from I-D to I-E and IV-B to IV-C have the same effect. The decreasing trend in $\overline{\lambda_1}$ during branching is seen best in series III. Similarly, from I-A to I-B and I-C, a decreasing trend is present during cavity formation. When cavities turn into holes, the decreasing $\overline{\lambda_1}$ trend continues, for example in I-D. It is also observed from series IV that the decrease in the number of holes increases $\overline{\lambda_1}$. From series VI, it is observed that increasing details decreases $\overline{\lambda_1}$. Lastly, disconnecting objects decrease $\overline{\lambda_1}$ as seen in I-F and II-F.

We picked the worst results to show the effect of noise. For each series, the image with worst error is picked and the absolute error is plotted in Fig.7g for all noise conditions. From Fig.7g, it is observed that for $\sigma = \{1, 2\}$, $\overline{\lambda_1}$ change is mostly below 10%. For I-D, IV-A and VI-B, increasing the noise to $\sigma = 3$ still keeps $\overline{\lambda_1}$ within 10% error. In the case of extreme noise ($\sigma = 4$), for most cases $\overline{\lambda_1}$ changes drastically. CTC performs bad for series III where very thin branches exist. This is partly due to how we added the noise. Fig.6 shows the effect of this type of noise on the branches of III-D.

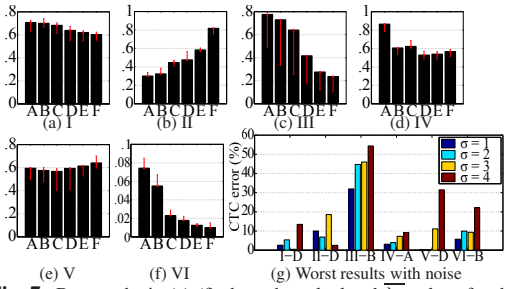
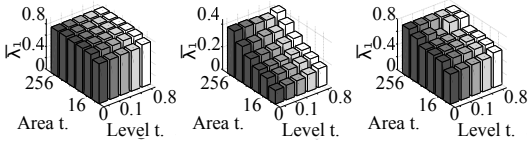


Fig. 7: Bar graphs in (a)-(f) show the calculated $\overline{\lambda_1}$ values for the 2D dataset. The error bars show the maximum and minimum values of $\overline{\lambda_1}$ in the presence of noise. Worst cases of error are shown in (g) for varying noise.

We picked one good, one bad and one typical case to show the effect of the pruning parameters on CTC. Fig.8 shows the changes for various combinations of level and area thresholds on images IV-A, III-D and V-A. It is observed that pruning does not alter $\overline{\lambda_1}$ significantly for image IV-A despite the noise. Whereas for III-D, pruning parameters affect the $\overline{\lambda_1}$ value. On a typical case, pruning parameters do alter $\overline{\lambda_1}$. However, it can be observed from Fig.8 that there is a flat region in a typical case which makes it easy to decide on the parameters. For example, for all results in this paper, a level threshold of 0.1 and an area threshold of 16 is used regardless of the data or the noise.



(a) Good case: IV-A with noise $\sigma = 4$ (b) Bad case: III-D with noise $\sigma = 4$ (b) Typical case: V-A with noise $\sigma = 1$

Fig. 8: Bar graphs showing the effects of pruning parameters on (a) good (b) bad and (c) typical cases. As default values, area threshold of 16 and level threshold of 0.1 is found to work reasonable.

3.2. 3D test images

Euler number is the standard measure to study connectivity of microstructures of materials [13]. In order to show the complementary value of CTC, we created 6 groups of 3D samples that resemble microstructures. Each group contains 50 random images of dimensions $128\times128\times128$. Three of the groups are made of spheres (S1, S2, S3) and the rest are lines (L1, L2, L3). S1 and L1 have large objects, S3 and L3 have small. S2 and L2 have both sizes. All images have $\sim$1 foreground to background ratio. Samples are shown in Fig.9.

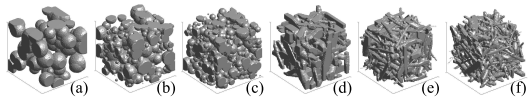


Fig. 9: 3D samples. (a) S1, (b) S2, (c) S3, (d) L1, (e) L2 and (f) L3.

Fig.10 shows the class conditional probabilities for Euler number and CTC fitted using Gaussian distribution. Samples with similar object dimensions have close $\overline{\lambda_1}$ values which is not observed for χ values. Based on the distributions, a naive Bayesian classifier yields a 72.3% classification accuracy using only χ. With $\overline{\lambda_1}$ included, the accuracy increases to 92.7%.

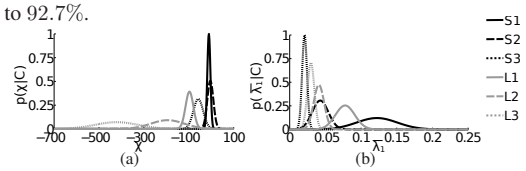


Fig. 10: Conditional probabilities for (a) Euler number (b) CTC.

The contour tree computation algorithm that is used in this work is $O(n \log n + M\alpha(M))$ (n vertices and M simplices) [10]. For the 2D and 3D test images, depending on the number of vertices in the contour tree, CTC calculation took a couple of seconds after the contour tree is computed.

4. CONCLUSIONS

In this article, we introduce a novel structural connectivity index which is based on contour trees and algebraic graph theory. CTC has interesting properties that complements the topological information provided by the Euler number. The applicability of our algorithm on any dimensions of data makes it a great tool for morphological analysis of binary images.

Appendix A: Proof of Theorem-1

We use the following corollary derived from Theorem 6.4.1 (page 240) and Corollary 6.4.3 (page 242) given in [14].

Corollary 3. *Suppose that $\hat{\mathcal{G}}$ is a subtree of the unweighted tree $\mathcal{G}$. Then $\lambda(\hat{\mathcal{G}}) > \lambda(\mathcal{G})$.*

Let $\mathcal{G}$ be any SCT. Pick $\hat{\mathcal{G}}$ as any path from any vertex on $\lfloor l^g_{max} \rceil$ to any vertex on $\lfloor l^g_{min} \rceil$. Note that $\mathcal{G}$ and $\hat{\mathcal{G}}$ have the same root vertex and height, h. Then $L(\hat{\mathcal{G}})$ can be expressed as the following $(h+1)\times(h+1)$ matrix:

$$L(\hat{\mathcal{G}}) = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}_{(h+1)\times(h+1)} \quad (2)$$

which has the eigenvalues $\lambda_k = 2(1 - \cos\frac{k\pi}{h+1})$, $k = \{0, 1, \cdots, h\}$ [15]. Finally, from Corollary 3, $\lambda_1(\mathcal{G}) \leq \lambda_1(\hat{\mathcal{G}}) = 2(1 - \cos\frac{\pi}{h+1})$.

A. REFERENCES

[1] M. H. F. Wilkinson, "Attribute-space connectivity and connected filters," *Image and Vision Computing*, vol. 25, no. 4, pp. 426–435, 2007.

[2] U. Braga-Neto and J. Goutsias, "A theoretical tour of connectivity in image processing and analysis," *Journal of Mathematical Imaging and Vision*, vol. 19, no. 1, pp. 5–31, 2003.

[3] M. H. F. Wilkinson, G. K. Ouzounis, and P. W. Hawkes, "Chapter 5 - advances in connectivity and connected attribute filters," in *Advances in Imaging and Electron Physics*, vol. 161, pp. 211–275. Elsevier, 2010.

[4] P. Salembier and M. H. F. Wilkinson, "Connected operators," *Signal Processing Magazine, IEEE*, vol. 26, no. 6, pp. 136 – 157, 2009.

[5] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Trans. on Image Proc*, vol. 9, pp. 860–872, 1998.

[6] J. K. Udupa and S. Samarasekera, "Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation," *Graphical Models And Image Processing*, vol. 58, no. 3, pp. 246–261, 1996.

[7] D. B. Aydogan and J. Hyttinen, "Binary image representation by contour trees," in *Proc.SPIE 8214, Medical Imaging 2012:Image Processing, 83142X*, Feb. 2012.

[8] Q.-Z. Ye, "The signed euclidean distance transform and its applications," in *Pattern Recognition, 1988., 9th International Conference on*, nov 1988, vol. 1, pp. 495–499.

[9] C. R. J. Maurer, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 2, pp. 265 – 270, 2003.

[10] H. Carr, J. Snoeyink, and U. Axen, "Computing contour trees in all dimensions," in *Proceedings of the eleventh annual symposium on Discrete algorithms*, San Francisco, California, United States, 2000, pp. 918–926, Society for Industrial and Applied Mathematic.

[11] H. Carr, *Topological Manipulation of Isosurfaces*, Ph.D. thesis, Department of Computer Science, The University of British Columbia, 2004.

[12] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 98, pp. 298–305, 1973.

[13] J. Ohser and K. Schladitz, *3D Images of Materials Structures: Processing and Analysis*, Wiley-VCH, 2009.

[14] J. J. Molitierno, *Applications of Combinatorial Matrix Theory to Laplacian Matrices of Graphs*, Taylor & Francis, 2012.

[15] W. C. Yueh, "Eigenvalues of several tridiagonal matrices," *Appl Math ENotes*, vol. 5, pp. 66–74, 2005.

PUBLICATION-III

Aydogan D.B., Moritz N., Aro H. & Hyttinen J.

Analysis of trabecular bone microstructure using contour tree connectivity

Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013, volume 8150, Springer, 428-435, Eds: Mori, Kensaku, Sakuma, Ichiro, Sato, Yoshinobu, Barillot, Christian, and Navab, Nassir, 2013

Analysis of trabecular bone microstructure using contour tree connectivity

D. Baran Aydogan[1,2], Niko Moritz[3], Hannu T. Aro[3], and Jari Hyttinen[1,2]

[1] Department of Electronics and Communications Engineering, Tampere University of Technology, Tampere, Finland
[2] BioMediTech, Tampere, Finland
[3] Orthopaedic Research Unit, Department of Orthopaedic Surgery and Traumatology, Turku University Hospital and University of Turku, Turku, Finland

Abstract. Millions of people worldwide suffer from fragility fractures, which cause significant morbidity, financial costs and even mortality. The gold standard to quantify structural properties of trabecular bone is based on the morphometric parameters obtained from μCT images of clinical bone biopsy specimens. The currently used image processing approaches are not able to fully explain the variation in bone strength. In this study, we introduce the contour tree connectivity (CTC) as a novel morphometric parameter to study trabecular bone quality. With CTC, we calculate a new connectivity measure for trabecular bone by using contour tree representation of binary images and algebraic graph theory. To test our approach, we use trabecular bone biopsies obtained from 55 female patients. We study the correlation of CTC with biomechanical test results as well as other morphometric parameters obtained from μCT. The results based on our dataset show that CTC is the 3^{rd} best predictive feature of ultimate bone strength after bone volume fraction and degree of anisotropy.

Keywords: trabecular bone, microstructure, binary morphology, CT imaging, connectivity, contour tree, algebraic graph theory

1 Introduction

Fractures affect millions of people worldwide, causing significant morbidity, financial costs and even mortality. Several studies suggest that currently used standard diagnosis approach which is based on BMD is a limited predictor of fracture risks [1]. On the other hand, there is increasing evidence on other factors, most important one being the bone quality, which combines both structural and material properties of trabecular bone tissue [2]. Currently, the gold standard to study trabecular bone microstructure is based on morphometric parameters obtained from μCT images. Therefore, it is crucial to develop effective image processing methods that predict bone strength by analyzing μCT images.

Connectivity has long been recognized as an important parameter to predict the mechanical properties of trabecular bone [3]. The most commonly used connectivity measure is the connectivity density (Conn.D), which is based on the

Euler number (χ) [4]. For a 3D binary image that is composed of a single connected component with no holes, $1 - \chi$ gives the number of tunnels that exist in the connected component. The tunnels in the case of trabecular bone are mostly associated with trabeculae. This makes Conn.D, which is defined as the number of tunnels per unit volume, an elegant tool to study trabecular bone topology. On the other hand, for most practical purposes, topology alone is not sufficient to describe connectivity properties. For example, although Conn.D includes information regarding the number of trabeculae, it does not provide information about the thickness of the connections nor the gaps between them.

In this study, we propose to use the contour tree connectivity (CTC) as an alternative connectivity measure to predict strength of trabecular bone. In [5], contour trees are utilized to provide graph representations of binary images. This representation of images transforms complex 3D data, such as the trabecular bone, into simple graphs. Based on a similar representation, CTC calculates the connectivity of binary images by calculating the connectivity of graphs.

This study introduces CTC as a novel morphometric parameter to study bone quality. For that, we study bone biopsies of 55 patients and show how CTC can be used to predict the strength of trabecular bone.

2 Methods and Materials

2.1 Contour tree connectivity (CTC)

Representation of binary images using contour trees: Contour tree is a topological abstraction that follows the evolution of level sets as they appear, join, split or disappear by means of a graph. In image processing, it has been mostly used for volume simplification and visualization of high dimensional scalar fields [6]. Contour trees are first used for binary images in [5], in order to develop a graph representation, which takes into account both the geometrical properties of foreground object(s) and how the object(s) is/are placed on the background. This is done by first applying the Euclidean distance transform (EDT) and than estimating the contour tree of the transformed image. An example on a small crop of a 2D trabecular bone slice is shown in Fig.1.

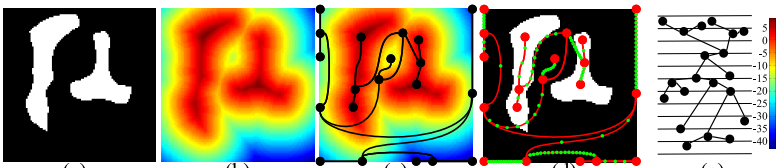


Fig. 1. (a) Input image (b) EDT of input (c) Contour tree drawn on top the EDT image (d) Supplemented contour tree with red nodes from the contour tree and green nodes from supplementation (e) Contour tree with level lines and EDT colormap.

Extraction steps and the algorithms used to compute CTC: The main idea of CTC is to compute the structural connectivity of a binary image by computing the connectivity of its contour tree representation. Fig.2 shows the flowchart for extracting the CTC feature.

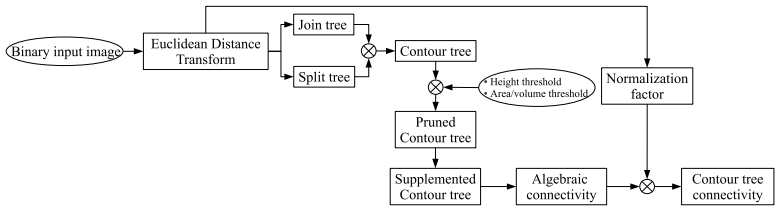


Fig. 2. Flowchart for the CTC feature extraction. From a binary input image, firstly EDT, then the supplemented contour tree (SCT) and finally the CTC is calculated as the normalized algebraic connectivity of SCT.

Starting with a binary image, we first create a real valued image using EDT. With a signed EDT, we transform the foreground to positive and background to negative real numbers with their distances to the nearest non-object pixels. The EDT image is used as the input for computing the contour tree.

We picked a popular method called the *sweep and merge* to compute the contour tree [7]. The sweep and merge method starts with computing the *join* and *split* trees; in a later stage, these are merged to form the contour tree. Join and splits trees represent the connectivities of upper and lower level sets respectively.

In order to remove the noise from the graph, we perform basic pruning operations on the contour tree. In order to prune edges or vertices, two user defined inputs are utilized. Edges to be removed must have less area/volume than the defined area/volume threshold and the vertices on each sides of the edge should have a level difference below the given level threshold.

As the last graph operation, we create a new tree by introducing integer valued levels in between the critical points. This final tree is called the supplemented contour tree (SCT). SCT simply shows how foreground/background regions in the image are positioned by the number of vertices along edges. The example in Fig. 1 shows the difference between the contour tree and the SCT.

Graph spectrum and algebraic connectivity: CTC measures the connectivity of a binary image indirectly by computing the connectivity of its contour tree representation. To achieve this, we use algebraic graph theory and spectrum of graphs. The spectrum of a graph $\mathcal{G}$ is the set of the eigenvalues of its Laplacian matrix that is given as $L(\mathcal{G}) = D(\mathcal{G}) - A(\mathcal{G})$. Here $D(\mathcal{G})$ and $A(\mathcal{G})$ are the degree diagonal and the adjacency matrices respectively. The eigenvalues of the Lapla-

cian matrix can be sorted as $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ for any connected graph. Since $\lambda_1 = 0$ for disconnected graphs, Fiedler named the second smallest eigenvalue, λ_1, as *algebraic connectivity*.

From the definition of algebraic connectivity, for all SCT, $\lambda_1 > 0$ is a lower bound. Also for all SCT, $\lambda_1 \leq 2(1 - \cos\frac{\pi}{h+1})$ is an upper bound, where h is the height of SCT. (The upper bound follows from Theorem 6.4.1 (page 240) and Corollary 6.4.3 (page 242) given in [8].)

Among the set of trees with same height, trees with wider horizontal spread of vertices have lower λ_1. Therefore, we define CTC of a binary image I by normalizing λ_1 with as follows:

$$CTC = \frac{\lambda_1(\mathcal{G})}{2\left[1 - \cos\left(\frac{\pi}{\lfloor l^g_{max} \rceil - \lfloor l^g_{min} \rceil + 1}\right)\right]} \tag{1}$$

Here $\mathcal{G}$ is the SCT of I. λ_1 is the algebraic connectivity of $\mathcal{G}$. l^g_{max} and l^g_{min} are the global max and min of EDT of I and $\lfloor . \rceil$ is the rounding operator.

From Equation 1, it follows that for any binary image, $0 < CTC \leq 1$.

2.2 Patient data, μCT imaging and biomechanical testing

The original study population consisted of 61 patients with primary hip osteoarthritis who underwent primary cementless total hip arthroplasty (THA). The demographic data of the patients was described previously in [9]. During THA surgery, a cancellous biopsy was taken from the intertrochanteric region of the proximal femur as described earlier in [10]. The biopsy specimen were wrapped in saline-wet sponges and placed in a sealed plastic bag and stored frozen at -20°C until machining. Multiple specimens were prepared from each biopsy under saline irrigation using a high-speed trephine drill. The cylindrical specimens were scanned with μCT (SkyScan 1072, Kontich, Belgium) with isotropic voxel resolution of 16.3 μm.

From the μCT images, 10 parameters were calculated. These are: bone volume fraction (BV/TV), bone mineral density (BMD), mean trabecular thickness (Tb.Th), trabecular seperation (Tb.Sp), trabecular number (Tb.N), trabecular bone pattern factor (Tb.Pf), structure model index (SMI), degree of anisotropy (DA), Conn.D and CTC. Conn.D and CTC were calculated using MATLAB, whereas the other parameters were calculated using CTan software (SkyScan). For each patient, the mean results of the parameters obtained from multiple specimens were used in the study.

Following the μCT imaging, the specimens were used in compression testing to failure using a universal mechanical testing device (Avalon Technologies, Rochester, MI, USA). Tests were performed at a constant speed of 0.825 mm/min and the load data were continuously recorded by the data acquisition system (Visual Designer, Intelligent Instrumentation, Tucson, AZ, USA). Values for the ultimate tensile strength (σ_U) were calculated from the raw data files using Origin software (Origin Lab Corp. Northampton, MA, USA).

The analyses were performed on 55 patients out of 61. Six patients were excluded due to incomplete data. The patients were divided into three groups with respect to their trabecular bones' ultimate strength. The first group, G-1, has $\sigma_U \leq 0.35$ MPa, the second group, G-2, has 0.35 MPa $< \sigma_U \leq 1$ MPa and the last group, G-3, has $\sigma_U > 1$ MPa. The thresholds were chosen manually by taking into account the jumps in σ_U. According to this division, 26 patients belong to G-1, 20 patients belong to G-2 and 9 patients belong to G-3. Fig.3 shows the values for the ultimate tensile strength in ascending order and the threshold levels used to group the patients.

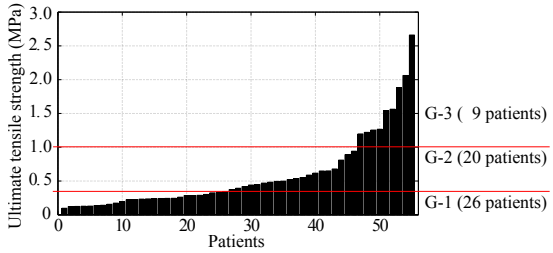


Fig. 3. Ultimate strength values obtained from the measurements in ascending order. Red lines show the thresholds that are used to categorize the patients into three groups.

2.3 Regression analysis, feature selection and classification method

The correlation between ultimate tensile strength and BV/TV was reported earlier in [10]. In this work, we study the other correlations and calculate the coefficient of determination (R^2) between *(a)* σ_U and μCT parameters *(b)* BV/TV and μCT parameters *(c)* CTC and μCT parameters.

To identify the most predictive μCT parameters for ultimate tensile strength, we used the data from all the patients and did a sequential forward feature selection (SFS). SFS selects a subset of μCT features that best predict the σ_U measurements by sequentially adding features. Each feature is selected after a repeated application of 10-fold cross validation. Among all candidates, the feature set that gives the minimum mean deviance is picked. Deviance measures the quality of the fit as the residual sum of squares obtained from generalized linear model regression. We used SFS on two sets of features. The first feature set, Case-1, had all the 10 μCT parameters. The second feature set, Case-2, did not have CTC but had all the other 9 parameters.

Lastly, we used linear discriminant analysis for classifications. For this, we used 13, 10 and 4 patients from groups G-1, G-2 and G-3 respectively as the training set. The rest of the 28 patients are used in the tests. For both feature sets, Case-1 and Case-2, we sequentially classified test subjects into groups by increasing the number of significant features.

3 Results

3.1 Demonstrative CTC study

We picked three μCT images to show example CTC values for different cases. Fig.4 shows these samples, corresponding ultimate tensile strength, BV/TV and CTC values. Since the actual contour trees contain several hundreds of vertices, simplified versions obtained from the crops are plotted below each μCT image.

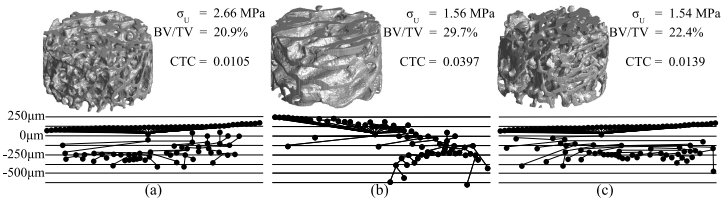


Fig. 4. Example μCT images and corresponding σ_U, BV/TV and CTC values. Below each sample is a simplified version of the contour tree. (a) shows the strongest bone in the dataset, (b) has the highest BV/TV in the dataset, (c) has a similar σ_U value to the one in (b) despite less BV/TV.

3.2 Correlation analysis

Fig.5 shows the calculated R^2 values. From left to right, Fig.5 shows the linear correlation between (a) σ_U and μCT parameters (b) BV/TV and μCT parameters (c) CTC and μCT parameters.

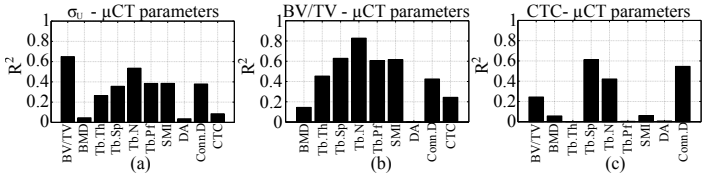


Fig. 5. Linear regression analyses show (a) σ_U is best correlated with BV/TV (b) Tb.N, Tb.Sp, SMI and Tb.Pf are correlated with BV/TV (c) CTC is uncorrelated with μCT parameters other than Tb.Sp, Conn.D and Tb.N.

3.3 Feature selection and classification results

Table 1 shows the order of the selected features for Case-1 and Case-2 using SFS and the classification accuracies.

Table 1. The order of the selected features that best predict σ_U measurements and the classification accuracies using feature sets Case-1 and Case-2. CTC is chosen as the 3^{rd} best feature by SFS. BV/TV and DA are the top two features.

	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
Case-1	BV/TV	DA	CTC	Conn.D	Tb.Sp	Tb.Pf	Tb.Th	Tb.N	SMI	BMD
	60.7%	50%	64.3%	67.9%	67.9%	67.9%	71.4%	71.4%	71.4%	67.9%
Case-2	BV/TV	DA	SMI	Conn. D	Tb.N	Tb.Th	Tb.Sp	BMD	Tb.Pf	-
	60.7%	50%	60.7%	60.7%	60.7%	53.6%	53.6%	53.6%	57.1%	-

4 Discussion and conclusion

Traditionally, the connectivity of trabecular bone has been studied from a topological point of view using the Euler number. This interpretation has been linked to the mechanical properties of bone [4]. With a structural interpretation, CTC provides new connectivity information that takes into account the thickness of the trabeculae as well as their separation.

Fig.4 shows example CTC values obtained for 3 μCT images. Generally an increased number of trabeculae decreases CTC by two mechanisms. Firstly, trabeculae appear as local maximas in the EDT image, which introduces vertices into the join tree. Secondly, gaps between trabeculae, that appear as local minimas, introduce vertices into the split tree. As a result, a sample with a large number of trabaculae is represented with a spread of several vertices above and below 0 level. In the contour trees shown in Fig.4, vertices at higher levels correspond to thicker trabeculae, whereas lower levels correspond to larger gaps between trabeculae. By quantifying the connectivity of this representation, CTC quantifies the connectivity of trabeculae with regard to their thickness and separation.

Fig.4 also shows how CTC complements other features such as BV/TV. From Fig.4a to Fig.4b, there is a 1.1 MPa decrease in σ_U despite the 8.8% increase in BV/TV. Additionally, from Fig.4b to Fig.4c, the 7.3% decrease in BV/TV does not change σ_U significantly. The contour tree for Fig.4b with few trabeculae and large gaps accounts for these differences. This compact tree representation yields a large CTC value, which indicates structural weakness for this sample.

The correlation analysis in Fig.5a shows that BV/TV alone explains more than 65% of the variance in σ_U. Second most correlated feature with σ_U is Tb.N. Fig.5b shows that, Tb.N, Tb.Sp, SMI and Tb.Pf are correlated with BV/TV. On the other hand, CTC is uncorrelated with most parameters.

Table 1 shows that CTC is the 3^{rd} best predictive feature of ultimate strength after BV/TV and DA. This coincides with both earlier and recent studies where BV/TV and DA are shown to be good predictors for σ_U [11, 12]. Interestingly, our results show that BMD is neither correlated with σ_U nor it is significant when predicting σ_U. It is observed from the correlation and classification results that most of the commonly used parameters are redundant. For Case-2, once BV/TV is used, the inclusion of many features does not improve accuracy. Moreover, additional features decrease the accuracy more for Case-2 than Case-1 due to

the curse of dimensionality. Lastly, it is observed that 3 subsets of Case-1 that include CTC reach an accuracy of 71.4%, which could not be achieved by any Case-2 subsets.

The application of linear regression for correlation analysis has the limitation that only linear relations are investigated. LDA also suffers from the same problem regardless of our use of independent traning and test sets. Further investigations that include non-linear relations might improve the results. Additionally, in order to generalize the results, a wider spectrum of patients need to be studied.

In this article, we introduce the contour tree connectivity (CTC) to predict the ultimate strength of trabecular bone. Our results show that most morphometric parameters do not provide additional information on top of bone volume fraction. CTC on the other hand, is uncorrelated with bone volume fraction and most other parameters. Our findings show that CTC is the 3^{rd} best predictive parameter for ultimate strength after bone volume fraction and degree of anisotropy. By providing new structural connectivity information, CTC offers more options to better predict bone quality.

References

1. Heaney, R.: Is the paradigm shifting? Bone **33**(4) (2003) 457–465
2. Bouxsein, M.L.: Bone quality: where do we go from here? Osteoporosis International **14** (2003) 118–127
3. Fields, A., Keaveny, T.: Trabecular architecture and vertebral fragility in osteoporosis. Current Osteoporosis Reports **10**(2) (2012) 132–140
4. Odgaard, A., Gundersen, H.J.G.: Quantification of connectivity in cancellous bone, with special emphasis on 3-d reconstructions. Bone **14**(2) (1993) 173–182
5. Aydogan, D.B., Hyttinen, J.: Binary image representation by contour trees. In: Proc.SPIE 8214, Medical Imaging 2012:Image Processing, 83142X. (Feb 2012)
6. Carr, H., Snoeyink, J., Van De Panne, M.: Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. Computational Geometry: Theory and Applications **43**(1) (2010) 42–58
7. Carr, H.: Topological Manipulation of Isosurfaces. PhD thesis, Department of Computer Science, The University of British Columbia (2004)
8. Molitierno, J.J.: Applications of Combinatorial Matrix Theory to Laplacian Matrices of Graphs. Taylor & Francis (2012)
9. Mäkinen, T.J., Alm, J.J., Laine, H., Svedström, E., Aro, H.T.: The incidence of osteopenia and osteoporosis in women with hip osteoarthritis scheduled for cementless total joint replacement. Bone **40**(4) (2007) 1041–1047
10. Moritz, N., Alm, J.J., Lankinen, P., Mäkinen, T.J., Mattila, K., Aro, H.T.: Quality of intertrochanteric cancellous bone as predictor of femoral stem rsa migration in cementless total hip arthroplasty. Journal of Biomechanics **44**(2) (2011) 221–227
11. Goulet, R.W., Goldstein, S.A., Ciarelli, M.J., Kuhn, J.L., Brown, M.B., Feldkamp, L.A.: The relationship between the structural and orthogonal compressive properties of trabecular bone. Journal of Biomechanics **27**(4) (1994) 375–389
12. Tassani, S., Öhman, C., Baleani, M., Baruffaldi, F., Viceconti, M.: Anisotropy and inhomogeneity of the trabecular structure can describe the mechanical strength of osteoarthritic cancellous bone. Journal of Biomechanics **43**(6) (2010) 1160–1166

PUBLICATION-IV

Aydogan D.B., & Hyttinen J.

Characterization of microstructures using contour tree connectivity for fluid flow analysis

J R Soc Interface 2014 11: 20131042

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3314-3
ISSN 1459-2045