A Critical Examination of Two Specific Approaches Used to Characterize Medical Images:

i) Projection-based Descriptors for Image Retrieval and ii) Estimating Fractal Dimensions of Discrete Sets

by

Alison K. Cheeseman

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Giorgio Mantica Professor, Dipartimento di Scienza ed Alta Tecnologia University of Insubria, Como (Italy)
Supervisor(s):	Edward R. Vrscay Professor, Dept. of Applied Mathematics University of Waterloo
Internal Member:	Giang Tran Assistant Professor, Dept. of Applied Mathematics University of Waterloo
Internal Member:	Mohammad Kohandel Professor, Dept. of Applied Mathematics University of Waterloo
Internal-External Member:	Alexander Wong Professor, Dept. of Systems Design Engineering University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Alison Cheeseman was the sole author of Chapter 1, which was not written for publication. The remaining content in Parts I and II contains some material which was written for publication. The relevant contributions are as follows.

Alison Cheeseman is the sole author of Chapter 2. Chapters 3 and 4 contain select material and results which were previously written for publication in [1] and [2]. This research was conducted by Alison Cheeseman under the co-supervision of Dr. Edward R. Vrscay and Dr. Hamid R. Tizhoosh. Alison Cheeseman wrote both draft manuscripts, which all co-authors provided intellectual input on.

Alison Cheeseman is the sole author of Chapters 5 and 6. Chapters 7 and 8 contain material from [3] with additional details and some additional results in Section 7.3.3 which were added for this thesis. This research was carried out primarily by Alison Cheeseman under the supervision and guidance of Dr. Edward R. Vrscay. The manuscript in [3] was primarily written by Alison Cheeseman, with input from Dr. Vrscay. Appendix A was originally written for and published in [3] and was written by Dr. Vrscay with input from Alison Cheeseman.

Abstract

In this thesis we provide a critical examination of two methods which are used to characterize medical images. Accordingly, this thesis is split into two main parts. First, we take a look at the problem of designing efficient, compact image descriptors for content-based image retrieval of digital histopathology slides. Our approach here is twofold, in that we propose a frequency-based approach to encoding projection gradients and we study the effect of separating histology slides into two colour components based on a typical staining protocol. Our frequency-based approach is shown to be more effective in terms of search performance and efficiency than the standard MinMax method of binary encoding often employed in the literature. Furthermore, we find that by separating histopathology images into their stain components, we see a significant improvement in search accuracy over the use of greyscale images, and comparable, and often superior performance to the use of three channel RGB colour images as inputs. The results in this part of the thesis not only stand on their own as a solution for image search, they can also be applied to improve the efficiency and performance of future research in this field.

In the second part of this thesis, we consider the use of fractal dimensions as a method to characterize vascular networks, and other branching structures such as streams, and trees. We discuss the self-similarity (or lack thereof) of branching structures, and provide a clear argument against the use of the typical methods, such as the box-counting and sandbox methods, to estimate fractal dimensions from finite images of branching networks. Additionally, local slopes are used as a tool to illustrate the issues with these approaches when they are applied to branching structures, such as computer-generated fractal trees and retinal vascular networks. Some alternative approaches are suggested which could be used for the characterization of complex branching structures, including vascular networks.

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Edward Vrscay, for providing both support and guidance throughout the many ups and downs we have encountered over the course of my degree. Thanks to Dr. Vrscay I have never felt like I was in this alone. I would also like to thank Dr. Hamid Tizhoosh for acting as my co-supervisor for a number of years and providing valuable guidance and insights during that time.

I would like to thank Drs. Giang Tran, Mohammad Kohandel, Alexander Wong and Giorgio Mantica for taking the time out of their busy schedules to sit on my examination committee. Their time and inputs are greatly appreciated.

I would like to specifically thank Dr. Mohammad Kohandel, along with Dr. Michelle Przedborski, for introducing me to the research dealing with fractal-based analysis of vasculature images as well as many valuable discussions during the early stages of this part of my research.

Sincere thanks must also go to Dr. Kenneth Falconer of the Mathematical Institute, University of St. Andrews in Scotland, for his very helpful correspondence during the course of my research relating to fractal dimensions and their estimation.

To my family and friends who have stuck by me throughout this long journey, thank you for providing much needed balance in my life.

Finally, I acknowledge the financial support which I received throughout the duration of my PhD. My research was supported in part by a Queen Elizabeth II Graduate Scholarship in Science and Technology and by the Natural Sciences and Engineering Council of Canada (NSERC) in the form of a Discovery Grant and a Canada Graduate Scholarship. The Faculty of Mathematics and Department of Applied Mathematics, at the University of Waterloo have also provided support in the form of teaching assistantships, for which I am grateful.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Due to significant advancements in image acquisition technologies over the last few decades, the study of medical images has rapidly made its way to the digital world. As the sheer volume of available data increases, computer-aided technologies for the analysis and characterization of medical images are becoming increasingly relevant in order to leverage all of the information which is now available [4]. In this thesis we provide an in-depth examination of two particular approaches which have been used for the characterization of medical images - the use of projection-based descriptors for content-based image retrieval in digital histopathology, and the use of finite box-counting methods to estimate the fractal dimensions of vascular networks and other naturally occurring branching structures. We point out here that these approaches, and their respective applications, are not related to each other, but are two disjoint topics which will be examined separately. As such, this thesis is comprised of two distinct parts, each of which could be considered as a standalone document in its own right.

In Part I we discuss the design of efficient image descriptors for the retrieval of similar histopathological images. Histology slides are one of the more recent imaging modalities to make the shift from analogue to digital, and they present some unique challenges which require new solutions. For example, unlike most other medical imaging modalities, histology slides are stained with dyes and contain colour information which is diagnostically relevant. It is important that this colour information is considered when characterizing histopathology images. In the pursuit of a compact and physically meaningful descriptor for histopathology images, we propose a frequency-based approach to encoding local projection data, combined with the separation of images into colour channels based on the stains present in the image. We show that the frequency-based approach is an improvement over the binary encoding method typically used in the literature. We also find that using stain separated image

components as inputs to the image search algorithm provides a significant improvement over using greyscale images. The use of stain separated images provides comparable, and often improved, results to using the standard RGB colour images as inputs, while also resulting in descriptors which are more compact, thus reducing the storage requirements and computation time needed for image search.

We note that the work described in this first part of the thesis corresponds to work which was carried out from approximately 2018-2020 and resulted in the publications [1] and [2]. In [1], the proposed approach was shown to be comparable, in terms of classification performance, to the state-of-the-art at the time of publication, achieving a balanced accuracy of approximately 80% on the publicly available IDC Kaggle data set [5]. That being said, the development of image search algorithms for digital histopathology is a rapidly advancing field. As better and larger data sets have become available, deep networks have become even better at classifying them. For instance, in [6], published in 2021, classification accuracies as high as 90% are reported for a number of tumour types using nearest neighbour search applied to features extracted from a fine-tuned deep network. In this thesis, instead of presenting the proposed approach as a complete and up-to-date solution for image search, we focus on the aspects of the work which remain relevant and are applicable to future developments in the field.

Part I is organized as follows. In Chapter 2 some relevant background information is presented pertaining to content-based image retrieval for digital histopathology. In Chapter 3 we present our two proposed approaches for the computation of compact image descriptors for digital histopathology. Finally, in Chapter 4 we include some experimental results on two publicly available histopathology data sets which demonstrate the effectiveness of our proposed approaches. In this chapter we also include a brief study on the impact of the choice of distance function used to search for similar images.

In Part II of this thesis we shift our focus to fractal dimensions and the methods which are used to estimate them from discrete images. The estimation of fractal dimensions has been used as a tool to characterize complex objects, including medical images, for many years. In particular, vascular networks with their hierarchical branching structure, are often thought to resemble fractal trees, and as such are classified based on estimates of their fractal dimensions and other fractal properties. In the literature, estimates of fractal dimensions have been found to distinguish between healthy and pathological vascular networks in a number of applications. In this work we present some words of caution regarding the use of the usual methods, such as the well-known box-counting method, to estimate fractal dimensions from branching structures (which includes vascular networks). This portion of the thesis describes work which was done recently, between the years 2020-2022, and culminated in the publication of [3].

Part II of this thesis is organized as follows. In Chapters 5 and 6 we present the necessary theoretical background on fractal sets, fractal dimensions and the methods which are used to estimate fractal dimensions. Specifically, in Chapter 5 we focus on fractal geometry and fractal sets, and in Chapter 6 we describe the many computational methods which exist for the estimation of fractal dimensions. In Chapter 7 we discuss specifically how branching structures, although they may be fractal, are not self-similar, and how this limits our ability to estimate their fractal dimensions using traditional approaches. Furthermore, we present evidence supporting this claim, including computational results using computer-generated images of fractal trees and real images of retinal vasculature. Chapter 8 concludes Part II, and concludes this thesis, with a discussion on some alternative approaches to characterizing the fractal properties of branching structures and suggestions on how research in this field should proceed.

Part I

A Compact Representation of Digital Histopathology Images via Frequency-Based Encoded Local Projections and Digital Stain Separation

Chapter 2

Content-based Image Retrieval (CBIR) for Digital Histopathology

Histopathology, the examination of tissue under a microscope to study biological structures as they relate to disease manifestation, has traditionally been carried out manually by pathologists working in a lab. It is only somewhat recently that the technology has advanced far enough that digitized histology slides have become widely available. With the widespread availability of digital histopatholgy images, there is now a pressing need for the development of automated technologies to relieve the workload on highly trained and specialized pathologists. For instance, approximately 80% of all prostate biopsies performed in the U.S. every year are benign, which suggests that prostate pathologists are likely spending a large portion of their time analyzing benign tissue. Benign tissue is typically easy to distinguish from cancerous tissue, therefore this represents a significant waste of time and resources that could be better spent analyzing patients with cancer, specifically focusing on cases where the disease is difficult to identify or classify [7, 8].

As it stands, pathology diagnosis is primarily based on the subjective opinion of highly educated pathologists. It has become increasingly clear that there is a need for quantitative image-based analysis of digital histopathology slides to improve both efficiency and accuracy. Quantitative assessments are important for both diagnostic purposes and in order to gain a better understanding into the underlying biological mechanisms of specific diagnoses. Consequently, computerized image analysis for digital histopathology has quickly become an active area of research. To date, researchers have developed a number of methods to accomplish specific tasks in pathological image analysis, such as image classification, image segmentation and the detection of various pathologies in images. With extensive training

on data sets which have been annotated by experts, computers are able to complete these tasks with good accuracy, however there is a hesitancy among the medical community to adopt such solutions in practice. In the medical field there is a responsibility to the patient first and a need to be able to justify each decision which is made. Therefore, it is understandable that the complete removal of pathologists from the diagnostic process is undesirable. Instead, it is preferable to consider how technology can be used to assist pathologists in their work. One approach to doing so which is quickly becoming popular is the development of systems which can efficiently and accurately retrieve similar images from a database.

Specifically, content-based image retrieval (CBIR) is the process of searching for images which share the same visual characteristics as a given query image. The identification and analysis of similar images can assist pathologists in quickly and accurately obtaining a diagnosis by providing a baseline for comparison. In particular, if the images in the database have been previously diagnosed, pathologists can refer to the diagnostic and treatment information of the retrieved images in order to guide their decision making [9]. This approach based on finding similar images is particularly effective for histopathology applications as it does not necessarily rely on extensive amounts of annotated data. The use of CBIR in digital histopathology, made possible by the ability to rapidly scan and store large quantities of digital histopathology slides, has the ability to revolutionize the landscape of pathology diagnosis for many years to come.

2.1 Digital Histopathology

It has been more than twenty years now since the first development of commercial whole-slide imaging (WSI) scanners which have the ability to digitize entire pathology glass slides [10]. In this time the technology has seen significant advances. Improvements in modern WSI technology allow for the rapid digitization and storage of entire pathology glass slides and as a result, digitized histopathology images are now widely available for both clinical and research purposes. Similar to other forms of medical imaging, digital histopathology has now become amenable to the application of computerized image analysis and machine learning methods [7, 8].

2.1.1 Whole Slide Imaging

The first WSI scanner was designed by James Bacus in 1994, however early WSI systems were expensive and slow, taking over 24 hours to scan a single slide [10]. Since then, the

technology has seen considerable improvements and there are now a number of commercial WSI scanners available on the market which are capable of automatically generating high-resolution scans of entire glass slides in minutes. Similarly, dramatic improvements in storage capabilities and computational power make it possible to process these large WSI datasets. With the recent FDA approval of WSI for primary diagnosis in the United States, such devices are quickly gaining popularity among medical practitioners [10].

Whole slide scanners have four main components: a light source, slide stage, objective lenses, and a high-resolution camera. Images of tissue sections are captured in a line-scanning fashion and digitally assembled to form an image of the entire slide. In order to reproduce the level of detail necessary for diagnostic and research applications, slides must be captured at sufficiently high resolution and magnification. In an optical system, both the magnification and resolution are determined by the properties of the objective lens used [11]. In a digital system, however, the the resolution of the camera sensor and the viewing monitor also have an effect. The resulting resolution of depends on the aforementioned device specific factors, however as an example, a typical whole slide image scanned at 40× magnification has a resolution of approximately 0.25 μm per pixel. This means that just a small one mm^2 area of slide consists of tens of millions of pixels, leading to extremely large file sizes when capturing entire slides [12]. For standard viewing and analysis of slides, scanning is typically done at 20× or 40× magnification, depending on the application. Even higher magnifications (60×, 80×, etc...) are available on some highly specialized scanners, but this is only used for very specific cases where such a high level of detail is needed [12]. Figure 2.1 shows some examples of image patches captured at 40× magnification. Each image patch is only a small portion of the entire slide and gives an indication of how much detail is present in these images.

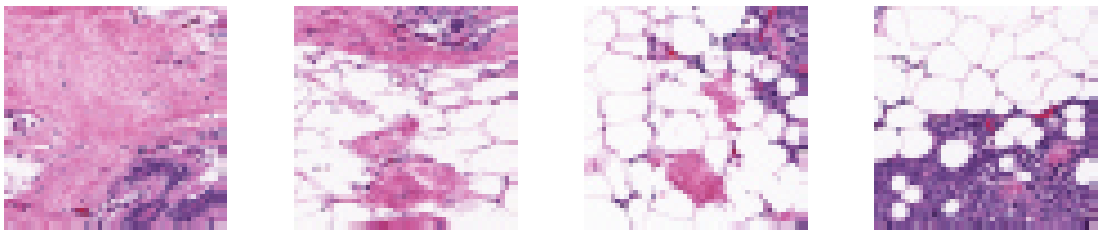

Figure 2.1: Four sample WSI image patches scanned at 40× magnification (from the publicly available invasive ductal carcinoma (IDC) Breast Histopathology dataset [5]).

2.1.2 Quantitative Image Analysis: Opportunities and Challenges

One of the main challenges in computer aided analysis of digital histopathology images is the enormous amount of data that algorithms must be able to deal with, compared to radiology and other imaging modalities. For example, some of the largest radiological data-sets are high-resolution chest CT scans which contain approximately $512 \times 512 \times 512$ spatial elements, more than 100 million voxels. On the other hand, a single prostate biopsy tissue sample digitized at $40\times$ resolution is composed of approximately 15,000 by 15,000 elements, that is, approximately 225 million pixels. To put this in context, a single prostate biopsy procedure contains anywhere between 12-20 biopsy samples, or approximately 2.5–4 billion pixels of data per patient study. Thus, unlike image analysis algorithms which have been used for radiology, algorithms for histopathological image analysis must be designed with efficiency in mind [7, 8].

In addition to large amounts of data, histopathological images present some other unique challenges. For one, while radiological images are grey-scale, histopathological images are typically stained with dyes to highlight certain features of the tissue. In order to properly use the relevant colour information, a histopathological image analysis system must be able to process colour images and determine relevant biological information from the presence of different stain colours. A whole slide pathology sample typically covers a fairly small region of the body at a very high resolution, meaning that the level of detail in these images is very high. As a result, histopathology images are thought to have more in common with textures than with radiological images which typically image larger regions of the body with more defined structure. A good example of this difference is seen in [13], where the results on three different data sets show that some traditional image retrieval methods developed for natural images perform much better on radiological images (x-ray and CT) than they do on the histopathological images used.

2.2 Content-based Image Retrieval

Content-based image retrieval, commonly denoted as CBIR, refers to the idea of using information contained within images to index and search through large image databases. The first notable research on CBIR occurred as far back as the early 1990s, and the field has seen considerable growth since then. Since the beginning, CBIR has been proposed for a wide variety of imaging applications, including weather forecasting, biological modelling, and of course medical imaging, the subject of this thesis [14]. The use of CBIR for specific medical imaging applications introduces some unique challenges, however there are also

some advantages. On the one hand, unlike commonplace images seen in the media which are easily understood by the average viewer, a medical image is likely difficult for an untrained viewer to make any sense of. This makes the development of CBIR algorithms difficult since this research is typically undertaken by image processing researchers, not medical practitioners. However, medical images are generally taken with a specific purpose in mind, and lack the contextual ambiguity that may be present in other readily available images. With medical images, it is also possible to take advantage of prior knowledge of anatomy and disease manifestation that exists within the medical community [15].

An image search, or CBIR, algorithm consists of two main steps: 1. the calculation of an image descriptor, and 2. the computation of a distance function or similarity measure between the query image and the remaining images in the database. An image descriptor is a mathematical description of the image that is used to capture relevant information in a compact representation that is suitable for comparing similarity between images. A good image descriptor should be designed carefully for the specific application at hand in order to accurately describe the relevant contents of the image, and may include information regarding the colour, texture, and shapes in an image, among other things. As we will discuss in Section 2.2.1, image descriptors may be handcrafted (constructed manually) or generated as the result of deep learning algorithms.

Given a collection of image descriptors corresponding to a set of images, the next step is to assess the similarity between the input query image and the images in the database in order to retrieve the most similar images to the query. This involves the computation of some sort of distance between pairs of image descriptors. We note that when dealing with images, the notion of distance (or similarity) is somewhat ambiguous. What makes two images similar? For instance, we might consider two images to be similar if they depict something similar, have many of the same pixels, or similar colours and patterns. Clearly, the way in which we define a distance between images is highly dependent on the context and a good CBIR algorithm will take this into consideration. In Section 2.2.2 we will discuss some of the different distance functions which can be used to assess the similarity of two images.

2.2.1 Image Descriptors

Very generally, an image descriptor is a quantitative description of an image which can be used for the purposes of sorting, classifying and/or characterizing imaging data in some way. A good image descriptor captures the essence of the image in an efficient manner. Typically, image descriptors are represented as vectors in $\mathbb{R}^n$ and are sometimes also referred to as

feature vectors. It is much easier, from a computational perspective, to compare feature vectors than to compare entire images. There are two main types of image descriptors: handcrafted descriptors and descriptors generated using deep learning methods, i.e. "deep features".

Handcrafted Descriptors

Image descriptors which result from the application of traditional image processing techniques, i.e. not deep learning, are referred to as handcrafted descriptors. Handcrafted descriptors have been used for both image classification and CBIR for many years [13]. A number of standard image descriptors exist which work well on most natural images, however medical images have unique properties and typically specific descriptors are tailored to particular applications. Although handcrafted descriptors have been surpassed in performance by deep features in many cases, trainable feature extraction methods such as deep networks are not always feasible in the medical field due to a lack of large balanced sets of labelled images. In addition, handcrafted descriptors provide a level of "explainability" which is not provided by deep features. In the medical field, where images are used to make diagnoses, this is important.

In general, handcrafted descriptors can be divided into two main types. The first are "keypoint" or "dictionary" descriptors. These are based on describing visual points of interest (keypoints) within the image. Some well known examples of keypoint descriptors include the surface invariant feature transform (SIFT) [16], speeded-up robust features (SURF) [17] and bag-of-visual words (BoVW) [18]. Keypoint descriptors have been found to be quite successful for image classification, including classification of histopathological images [19]. However, they require extensive training to generate the required set of keypoints (or the "dictionary") and are thus not that practical when dealing with large quantities of data.

The second major class of handcrafted descriptors are "histogram"-based descriptors. These are compact representations of images in the form of histograms, typically generated by counting local patterns. These histograms are usually normalized and can be thought of as the distribution of local patterns in the image. As a result, histogram descriptors are easily generalized to apply to images of different sizes without the need to crop or resize the images. Additionally, histogram descriptors do not require any training to generate, and are typically compact and easy to compute. Two very well known histogram descriptors are local binary patterns (LBP) [20] and histograms of oriented gradients (HOG) [21]. HOG was originally designed for the detection of humans in natural images, while LBP was originally designed for texture classification. Many variants of the LBP algorithm have been proposed and found to be highly effective in a number of applications, including medical

image classification [13]. In this thesis we will primarily be concerned with histogram descriptors due to their ease of computation and compact nature.

There are also some handcrafted descriptors which do not fall into either of these main categories. The GIST descriptor [22] is one such example. The GIST descriptor is a popular descriptor which is based on describing the "spatial envelope" of the scene. This is accomplished by convolving the image with a set of Gabor filters at different scales and orientations and constructing a vector based on the average values in local regions of each feature map. The GIST descriptor is generally considered to be a good descriptor for patterns and textures [23].

Deep Features

Although they will not be the focus of this work, it would be remiss not to touch on deep features, as they are increasingly becoming the standard for image search and classification in many fields. Deep features are image descriptors (feature vectors) generated from the intermediate steps of a deep learning algorithm. To generate deep features for a data set, a deep neural network, typically a convolutional neural network (CNN), is trained for image classification and high-dimensional feature vectors are extracted from the pooling or fully connected layers of the network. These types of methods have the potential to have very high discrimination power, but suffer from high dimensionality and the need for large volumes of balanced and labelled data to train the networks. When only a small labelled data set is available, as is often the case in histopathology, it is common to utilize networks which have been pre-trained on a larger data set (perhaps of natural images). To improve performance on a specific data set, the parameters of the pre-trained network can be fine-tuned using the data set of interest [13].

2.2.2 Distance Functions for Image Search

In order to search for and retrieve similar images, we need to define an appropriate distance function between image descriptors. A distance function is a function which outputs some measure of distance between elements of a set. The term distance function is often synonymous with distance metric (or just metric) - a distance function which satisfies some specific properties, such as the well-known triangle inequality. Distance metrics have many nice theoretical properties, however, in practical applications it is common to use distance functions which are not proper distance metrics, i.e. they don't satisfy all of the required properties. In this work we will use distance function to mean a function which assigns a distance to pairs of elements in a set, but which is not necessarily a distance metric.

In Chapter 4 we will present some experimental results in which an image search algorithm has been implemented with a number of different distance functions. Although we don't test every possible distance function (this would be impossible!), we test some of the most common distance functions, as well as introducing an additional distance function which is not typically used for image search, but has some desirable properties. Each of these functions is described in more detail below. Throughout the following we will consider $\boldsymbol{x}$ and $\boldsymbol{y}$ to be two descriptors of length n, i.e. $\boldsymbol{x} = [x_1, x_2, ..., x_n]$ and $\boldsymbol{y} = [y_1, y_2, ..., y_n]$. We will also assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are comprised of non-negative real numbers, i.e. $x_i, y_i \geq 0$ for all $i \in \{1, \ldots, n\}$.

The L_1 Distance

The L_1 distance between $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$d_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} |x_i - y_i|, \tag{2.1}$$

and goes by many names, including the Taxicab distance, Cityblock distance, and the Manhattan distance. These names are a reference to the grid layout of the streets in big cities such as Manhattan. The L_1 distance between two points on such a grid is the shortest distance that a taxi could take driving between them.

The Euclidean (L_2) Distance

The Euclidean (or L_2) distance is likely the most well-known distance function. In $\mathbb{R}^2$ and $\mathbb{R}^3$ the Euclidean distance between two points is the length of the shortest straight line which connects them. In the more general case of $\mathbb{R}^n$, the Euclidean distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as

$$d_2(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{n} (x_i - y_i)^2 \right)^{1/2}. \tag{2.2}$$

The Euclidean distance is the standard for measuring distances in Euclidean space.

The Chi-Squared Distance

The Chi-Squared (χ^2) distance comes from the χ^2 test in statistics. It is defined as [24]

$$d_{\chi^2}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2} \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{(x_i + y_i)}. \tag{2.3}$$

The division by $(x_i + y_i)$ in the sum weights differences between small values higher than differences between large values.

The Cosine Distance

The cosine similarity of $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined as [25]

$$sim(\boldsymbol{x}, \boldsymbol{y}) = \cos(\theta), \tag{2.4}$$

where θ is the angle between the two vectors. Clearly we have $-1 \leq sim(\boldsymbol{x}, \boldsymbol{y}) \leq 1$ where the maximum value indicates two vectors with an angle of zero between them (i.e. the most similar). In order to compute a distance based on the cosine similarity we define

$$d_{\cos}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \cos(\theta) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}. \tag{2.5}$$

The cosine distance defined in this way takes values from $0 \leq d_{\cos}(\boldsymbol{x}, \boldsymbol{y}) \leq 2$. The cosine distance is typically used when the magnitude of the vectors is not relevant (normalized histograms, for example).

The Correlation Distance

The correlation distance, as the name might suggest, is related to the correlation between the two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Specifically, the correlation distance is defined as [25]

$$d_{corr}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\sum_{i=1}^{n} (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{n} (x_i - \mu_x)^2} \sqrt{\sum_{i=1}^{n} (y_i - \mu_y)^2}}. \tag{2.6}$$

We note that this is very similar to the cosine distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, with the only difference being that we subtract the corresponding mean from each vector.

The Hutchinson Distance

Finally, we will also investigate the use of a distance between two probability measures (discrete or continuous) which is well-known in the mathematical literature, but which has received very little attention in imaging applications, and no attention to date in the field of digital histopathology. This distance, known by various names including the Monge-Kantorovich distance and Hutchinson distance [26], is considered to be a good measure of distance between measures or histograms. In [27] it is shown that in the finite one-dimensional case, the Hutchinson distance can be computed by summing the absolute value of the partial sums of the difference of the two measures, as follows

$$d_H(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{n-1} \mid \sum_{i=1}^{k} (x_i - y_i) \mid \tag{2.7}$$

where x and y are probability vectors of length n, i.e. $\sum_{i=1}^{n} x_i = 1$ and $\sum_{i=1}^{n} y_i = 1$. It is interesting to note that the right hand side of Equation (2.7) is the L_1 distance between the (discrete) cumulative distribution functions of the probability vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Although this does not extend to higher dimensions, this result is sufficient to allow us to compute the Hutchinson distance between one-dimensional image descriptors in linear time.

As an illustration of the benefits of the Hutchinson metric, we consider the simple example of two length n histograms each consisting of a single spike in different bins, j and k, as in $\boldsymbol{x} : x_i = 1$ if $i = j$, $x_i = 0$ elsewhere and $\boldsymbol{y} : y_i = 1$ if $i = k$, $y_i = 0$ elsewhere. Intuitively, we might expect that the distance between these two histograms should depend on the distance between the two spikes, however all of the distance metrics described above would assign the same distance to $\boldsymbol{x}$ and $\boldsymbol{y}$ independent of the values of j and k. Due to its cumulative nature, the Hutchinson distance is actually the exception to this rule, and we can show that $d_H(\boldsymbol{x}, \boldsymbol{y}) =\mid j - k \mid$. For certain image descriptors this may provide a better measure of distance than some of the more commonly used distance measures.

2.3 Literature Review

Research into algorithms for effective image retrieval dates back to before the 1990's, with the field really taking off in the late 1990's. Since then, an extraordinary number of papers have been published proposing CBIR algorithms for various applications [14]. In particular, medical imaging has often been cited as one of the main applications for image search technologies in terms of the potential impact on the field [28]. Initial applications within

the medical field focused on the use of CBIR to search for radiological images, such as X-ray and CT scans. Thanks to the introduction of WSI technology and digital histopathology in more recent years, extending these ideas to the more complex digital histopathology images was the next logical step. We note, as did the authors of [14], that the volume of research relating to CBIR over the years makes it unreasonable to provide a comprehensive review of the related literature in just a single document. Instead, in this Section, we present an overview of the more recent literature which served as the inspiration for our work. Specifically, we outline the research relating to CBIR for digital histopathology images ranging from around the mid-2010's to the beginning of 2019, when we began our study.

Due to the nature of the medical field, specifically, the need for "explainability" in any method used to come up with a diagnosis, and the lack of large balanced histopathology data sets which are available, there has been a significant amount of research into the design of handcrafted features for histopathology. For instance, in [29], a new method of image search based on graph-theoretic descriptions is proposed specifically for digital histopathology. While it is an interesting approach, this method suffers from issues of computational complexity. Many have also proposed modifications of traditional image descriptors for use on histopathological images, such as the hierarchical annular histogram described in [30] and the BoVW approach with multiple dictionaries proposed in [31]. In [13], a new histogram descriptor based on encoding local Radon projections in local neighbourhoods of images was proposed. This proposed Encoded Local Projections (ELP) image descriptor was found to be particularly successful on histopathology images. In fact, when applied to medical imaging applications, the ELP descriptor was shown to outperform many well-known handcrafted features, and even some deep features [13]. We will discuss the ELP descriptor in more detail in Chapter 3 as it has served as a starting point for our work.

Of course, there has also been significant study into the development of deep features for image search applications in histopathology. Although not the subject of this thesis, we briefly mention some of the related literature here in order to provide context for our work. One approach is to combine deep learning with more traditional approaches. For instance, in [32], the authors combine the classic BoVW approach with unsupervised feature learning to create a "bag of learned features". Others strictly use deep networks to generate features for image search, however the high-dimensionality of the resulting features is often a concern due to the so-called "curse of dimensionality". In [33] and [34], manifold learning techniques are used to learn lower dimensional representations of histopathology images for image search. Taking a different approach, in [35], deep features are binarized to enable faster searching. Due to the lack of large, balanced histopathological data sets, some researchers consider fine-tuning neural networks which have been pre-trained using some other data

set, for instance in [36] and [37].

Chapter 3

Two Approaches to Generating Efficient Handcrafted Descriptors

In this chapter we will take a closer look the process of generating efficient handcrafted descriptors for digital histopathology. Due to its success on histopathology images, we use the ELP image descriptor from [13], mentioned in the previous section, as a starting point for our work. As a result of our investigations, we propose two main innovations: the use of a frequency-based approach to encode projection gradients and the separation of the input images into physically meaningful colour channels. These two innovations stand on their own as a practical approach to CBIR for digital histopathology, however, this is a rapidly advancing field. We feel that the insights from our study are also applicable to the development of new approaches in the future.

3.1 A Frequency-based Approach

The basic idea behind the ELP method is to encode local projections of an image using a binary representation. A closer look reveals that the binary encoding used introduces some redundancy in the method, which our frequency-based approach aims to reduce. Before introducing our modified method, we first describe the ELP method in further detail. We note that in Section 3.1.1 we describe the ELP method as applied to greyscale images (where each pixel corresponds to a single intensity value). In Section 3.2 we describe how the ELP method (and our proposed method) can be applied in a similar manner to images with multiple colour channels.

3.1.1 Encoded Local Projections (ELP)

The ELP descriptor uses an image transform called the Radon transform to capture the shapes and patterns present in each image. The Radon transform is an integral transform which is well known in the image processing field - as its inverse transform is used to reconstruct images from CT scans. In addition to being used for the reconstruction of images, the Radon transform has found many other uses related to image processing and image analysis. It is particularly useful for detecting prominent lines/edges in images.

The Radon transform of a function $f(x, y)$ is formally defined as

$$R(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(\rho - x\cos\theta - y\sin\theta)dxdy, \tag{3.1}$$

where $\delta(\cdot)$ is the Dirac delta function. In the discrete case, as shown in Figure 3.1, the Radon transform consists of taking multiple projections of an image along parallel lines at varying angles θ. A projection in this case is defined as the sum of pixel values along a given line in the image. By taking many such projections we define a new two dimensional function of ρ and θ, $R(\rho, \theta)$, where $\rho = x\cos\theta + y\sin\theta$.

The goal of the ELP descriptor is to capture image attributes in local neighbourhoods. Instead of computing the Radon transform of the entire image, projections are computed over many small regions of the image. Consequently, the first step in the ELP algorithm is the identification of a set of small local windows for processing. In order to find projections which uniquely describe the patterns/textures in local neighbourhoods, only regions which are sufficiently non-homogeneous so as to ensure projections contain something of interest are considered. Letting $\boldsymbol{W}$ denote a local window of size $n \times n$, the homogeneity, H, of each window is calculated according to

$$H = 1 - \frac{1}{2^{n_{bits}}}\sqrt{\sum_i \sum_j (\boldsymbol{W}_{ij} - m)^2}, \tag{3.2}$$

where m denotes the median pixel value of $\boldsymbol{W}$ and n_{bits} is the number of bits used to encode the image. A threshold, T_H, ranging from $-\infty < T_H \leq 1$ is used to eliminate any windows with high homogeneity.

Once the relevant set of local windows has been defined, it is necessary to determine over which angles to compute projections. A good image descriptor for digital histopathology should be rotationally invariant, as the orientation of a scan is not diagnostically relevant. In pursuit of this goal, a unique angle is chosen in each window by which to "anchor" the

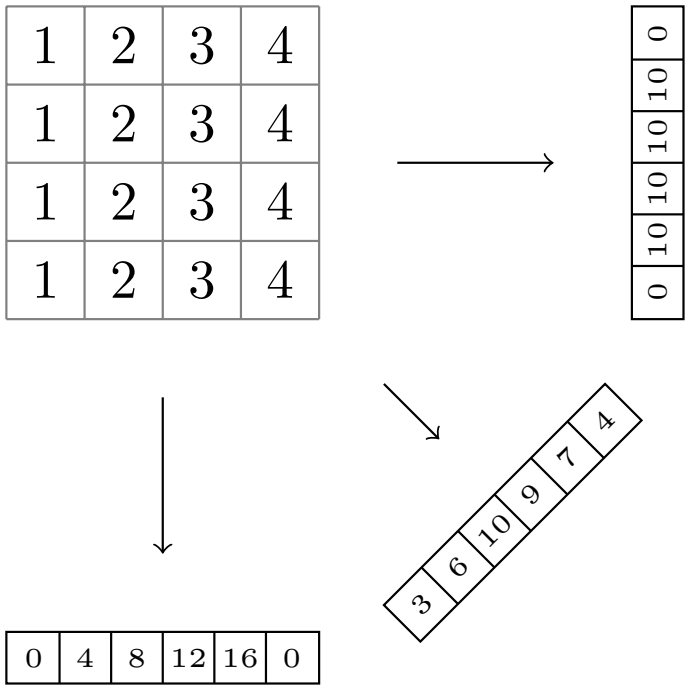

Figure 3.1: A simple illustration of the discrete Radon transform applied to a finite image. Zero-padding is added so that all projections are the same length.

projections. This anchor angle is chosen so as to find projections which capture significant changes in the window. In [13], the authors propose to search for an angle θ^*, along which the Radon projection, $\boldsymbol{p}$, has maximum amplitude over the entire Radon sinogram, i.e.

$$\theta^* = \text{argmax}_i \max \left[R(\rho_1, \theta_i), ..., R(\rho_{|\boldsymbol{p}|}, \theta_i) \right]. \tag{3.3}$$

For this fixed angle θ^*, the projection $\boldsymbol{p}_{\theta^*}$ is computed by taking the Radon transform along parallel lines ρ for the fixed anchor angle θ^*, and $\boldsymbol{p}_{\theta^*}$ is referred to as the anchor projection. Unfortunately, searching over many values of θ_i to find θ^* is prohibitively time-consuming. Instead, in [13], when applying their method to a histopathology data set, the authors use a more efficient method to approximate the anchor angle computation described above by taking θ^* as the median of the gradient directions present in the window. We also use this implementation of the ELP method in our work.

In order to encode the meaningful information from the Radon projection in a compact form, the authors choose to use "MinMax" encoding [38] applied to the gradient of the

projection. Given a projection vector $\boldsymbol{p}$ of length n, its derivative $\boldsymbol{p}'$, is approximated using a finite difference, i.e. $\boldsymbol{p}'(k) = \boldsymbol{p}(k+1) - \boldsymbol{p}(k)$. A binary encoding $\boldsymbol{b}$ is computed as

$$\boldsymbol{b}(i) = \begin{cases} 0 & \text{if } \boldsymbol{p}'(i+1) > \boldsymbol{p}'(i) \\ 1 & \text{otherwise} \end{cases} \tag{3.4}$$

for all $i \in \{1, 2, ..., n-2\}$. Following binary encoding of the anchor projection gradient, $\boldsymbol{b}$ is converted to an integer d, such that the histogram, $\boldsymbol{h}(d)$ can be incremented accordingly. Figure 3.2 shows an example of how the MinMax encoding scheme works, converting a projection gradient into a single integer value.

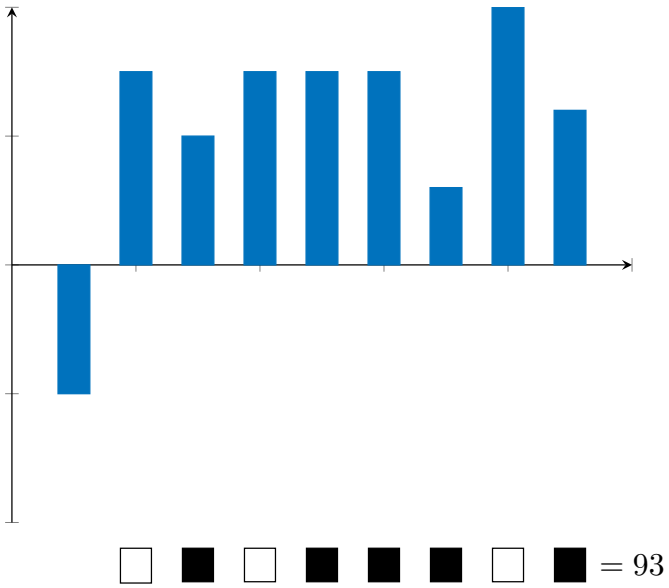

Figure 3.2: An illustration showing the use of the MinMax encoding scheme. A projection gradient (illustrated by a bar chart) is converted into a corresponding binary value. Empty boxes indicate a zero in the binary encoding and black boxes indicate a one.

To collect more information from each spatial window, three additional projections anchored to θ^* are computed at equidistant angles given by: $\Theta = \left\{\theta^*, \theta^* + \frac{\pi}{4}, \theta^* + \frac{\pi}{2}, \theta^* + \frac{3\pi}{4}\right\}$. Each projection is encoded, converted to a decimal and counted in the same manner. A final histogram can then be generated in two ways, merging the histograms of all binarized derivatives of projections along Θ into one histogram, or concatenating all four histograms into one long histogram. In [13] the method of concatenation, detached histograms, is found to have a slightly higher discrimination power, so this is what we consider for our work.

3.1.2 Frequency-based Encoded Local Projections (F-ELP)

As was previously mentioned, the ELP descriptor has had considerable success when applied to digital histopathology data sets. That being said, the method does have some limitations. In particular, the histograms which result from the binary encoding of the projections are quite long. Instead of storing entire projections, as in the ELP method, we propose to quantify the number of changes in each projection gradient and use this as an estimate of local frequency. As a result, we refer to our method as the Frequency-based Encoded Local Projections (F-ELP) method. While the original ELP method results in large histograms, the size of our F-ELP histograms depends linearly on the local window size, which is typically quite small. The compact nature of our descriptor is desirable from the perspective of both memory usage for storage of descriptors and computation requirements when applied to image retrieval and classification type tasks. The computation of our proposed F-ELP descriptor follows the same overall steps as the ELP descriptor, with some modifications along the way to improve rotational invariance, reduce sensitivity to shifts in the image and reduce redundancy by encoding only the frequency information from each projection. In the following, we go through each step of the algorithm, highlighting how our method differs from the ELP method.

Identify local windows

Here, our method does not differ at all from the original ELP method. We choose a threshold, T_H, and select all windows which are sufficiently non-homogeneous to be included in our computation of the descriptor.

Determine the anchor angle, θ^*

Similar to the ELP method, we seek a unique angle in each window by which to anchor our projections. We do so by computing the image gradient, binning the gradient directions into one degree intervals and selecting θ^* to be the mode (most frequently occurring) of the gradient directions. Our approach differs just slightly from the original method, in that we choose to use the mode instead of the median to find the average angle. We do so as the median is not invariant under circular shifts (i.e. angular rotations), whereas the mode is invariant, so long as there is one unique angle which occurs at the highest frequency (i.e. there exists a clearly dominant direction in the window).

Compute the projection along θ^*

As in the ELP method, we compute projections $\boldsymbol{p}_{\theta^*}$ by taking the Radon transform along parallel lines ρ for the fixed anchor angle θ^*. We also compute three additional equidistant projections anchored to $\boldsymbol{p}_{\theta^*}$.

Encode projections and create histogram

It is in the encoding of the projections where our algorithm differs the most notably from the ELP method. Instead of encoding the entire gradient of each projection, we quantify the gradient changes in the projection vector and use this to build our histogram. The benefits of this modification are two-fold. Primarily, we remove the storage overhead of encoding entire projections, and instead just capture the general trend (low or high frequency) of the projections along each direction, resulting in much smaller histograms which still perform very well. Our proposed method also avoids the use of a binary encoding to capture the projections. This is beneficial as the binary encoding used by the ELP is very sensitive to small shifts in the projection, i.e. a change in one binary bit can lead to a very large difference in the resulting histogram. On the other hand, when the local projection frequency changes, the resulting change in the F-ELP histogram reflects the size of the frequency change.

Given a projection vector $\boldsymbol{p}$ of length n and its derivative $\boldsymbol{p}'$, we compute the following quantized encoding of the derivative,

$$\boldsymbol{q}(i) = \begin{cases} 0 & \text{if } \boldsymbol{p}'(i) \leq -T \\ 1 & \text{if } |\boldsymbol{p}'(i)| < T \\ 2 & \text{if } \boldsymbol{p}'(i) \geq T. \end{cases} \tag{3.5}$$

The three levels given in Eq. (3.5) indicate regions where the projection, $\boldsymbol{p}$, is decreasing, nearly constant (we use a small threshold, T, here to ignore small fluctuations), and increasing, respectively. Next, we count the number of transitions in $\boldsymbol{q}$ to get our estimate of local frequency which will be an integer value, d which satisfies $0 \leq d \leq n-2$. Once we have d we can increment the histogram $\boldsymbol{h}(d)$. Similar to the original ELP descriptor, we obtain more information by computing three additional equidistant projections relative to our anchor angle θ^* and concatenate the results to generate the final histogram.

Figure 3.3 shows an example of both the ELP and F-ELP descriptors computed for two sample images from the IDC dataset which clearly depict visually distinct samples. In order to compute the pictured ELP and F-ELP descriptors a window size of $n = 9$ was

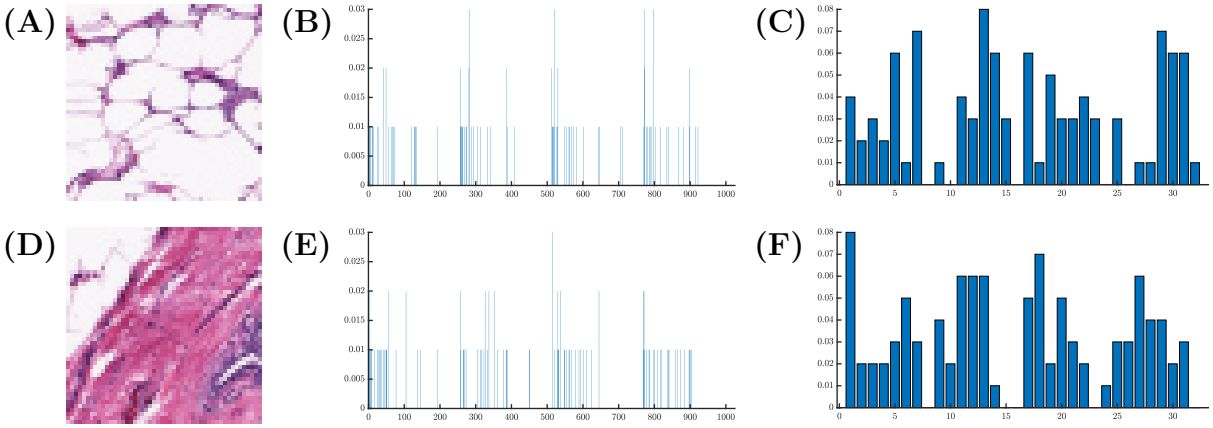

Figure 3.3: Sample histograms generated using the ELP and F-ELP methods with a window size of $n = 9$. In (A) we show an image patch from the IDC data set, with (B) and (C) showing the resulting ELP and F-ELP histograms, respectively. Similarly, in (D) we have another image patch, with corresponding ELP and F-ELP histograms in (E) and (F).

used and the input images were converted to greyscale, resulting in an ELP descriptor of length 1024 and an F-ELP descriptor of length 32. Both histograms have been normalized according to the L_1-norm. From the figure we can see that the F-ELP descriptor, although it has fewer bins, appears to show a more varied distribution. When looking at the ELP histograms, we observe that the distribution is similar for both images, with differences occurring in only a few bins. This indicates that there is some redundancy in this image representation which is what we try to remove with the F-ELP method.

3.2 Digital Stain Separation

In the previous sections we discussed the application of the ELP and F-ELP descriptors to greyscale (or intensity) images. Histopathology images, as shown in Figure 2.1, are generally colour images. It is, of course, possible to convert colour images to their greyscale intensity values and apply the methods directly as described above (as is done in [13]), however in doing so we are leaving out some important information. Prior to imaging, histology slides are stained to enhance the detail in tissues and cells, and thus the colours present in an image are biologically meaningful. The obvious solution might be to incorporate the colour information by computing a descriptor for each of the RGB colour channels, however this doesn't account for the variation in stain colouring which results from inconsistencies

in tissue preparation, sample thickness and scanning procedures [39]. The most common staining protocol used to stain histology slide in practice involves two chemical stains; hematoxylin and eosin (H&E). Hematoxylin is a chemical which stains cell nuclei blue, and eosin stains other structures varying shades of red and pink [40]. The amount of each of these colours which appears in a slide and the size, shape and frequency at which they appear are all relevant factors a pathologist might assess when making a diagnosis. For this reason, we consider separating the input images into two colour components which correspond to the amount of each stain present at each pixel. By working directly with the hematoxylin and eosin components, we are able to adjust for differences in stain colours across images from different sources.

A number of methods for the digital stain separation of H&E slides which perform quite well already exist in the literature. Although their intended usage is for stain normalization to control for variation in stain intensities and colours, these same methods are suitable for our purposes. In this work we adopt the method proposed in [41], an extension of the wedge finding method from [42]. Unlike some other methods which have been proposed for stain separation of histopathology images [40] the proposed method does not require any calibration or knowledge of the exact stain colours. Instead, it works by using the available image data to estimate an H&E colour basis. Given that our method should ultimately be applicable to data arising from multiple sources, this is an important feature of the stain separation algorithm.

Figure 3.4 shows the results of applying the stain separation algorithm to a sample image patch from the IDC data set. We can see from the original image in (A) how two distinct structures are highlighted by the two stains. The images in (B) and (C) show the

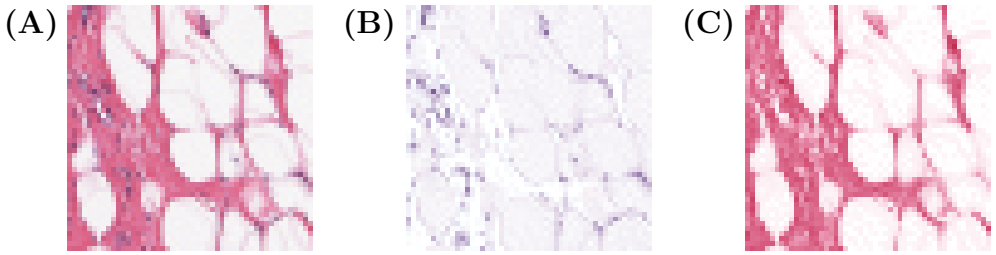

Figure 3.4: An illustration of the stain separation process. In (A) we show the original H&E stained image, in (B) the separated hematoxylin component, and in (C) the eosin component.

separated hematoxylin (blue/purple) and eosin (pink/red) components, respectively. These images clearly illustrate the ability of the stain separation algorithm to accurately isolate both components. Similar results were observed on other images in the data set.

Given the resulting stain separated image components, we proceed as described in the previous sections to compute an image descriptor for each component of of the image separately. Since we are using two colour channels, this results in two histograms, $\boldsymbol{h}_H$ and $\boldsymbol{h}_E$ which are concatenated to form the final longer histogram $\boldsymbol{h} = [\boldsymbol{h}_H \ \boldsymbol{h}_E]$. We note that this method can also be applied to generate histograms with any number of input colour channels, where the length of the final histogram is obviously dependent on the number of colour channels used. For instance, the standard three RGB colour channels can be used as inputs and the resulting histograms concatenated to form $\boldsymbol{h} = [\boldsymbol{h}_R \ \boldsymbol{h}_G \ \boldsymbol{h}_B]$. Later, in Section 4.3, we will compare histograms generated using both the aforementioned H&E colour channels and the standard RGB colour channels.

Chapter 4

Experimental Results

In this chapter we will present some experimental results evaluating the effectiveness of our proposed approaches. In order to clearly understand their effects, we will test three different aspects of the image search process in isolation. First, in Section 4.2, we evaluate our frequency-based approach to encoding projections by direct comparison of the F-ELP descriptor with its predecessor, the ELP descriptor. In Section 4.3 we study the effectiveness of separating the input images into their H&E colour channels, as compared to using greyscale or RGB images as input. This comparison is done using some well-known image descriptors, as well as the ELP and F-ELP descriptors specifically for digital histopathology. Finally, in Section 4.4 we look at how the choice of distance function used for image search impacts the results.

4.1 Implementation Details

In order to evaluate the performance of our proposed approaches we have implemented our proposed descriptor along with some additional descriptors for comparison. We have also implemented the digital stain separation algorithm using Matlab code from [41]. Two publicly available annotated histopathology data sets were used to evaluate image search performance. The relevant details of our implementation are described in the following.

4.1.1 Data Sets & Image Preprocessing

For this study we use the invasive ductal carcinoma (IDC) data set from *Kaggle*, a popular online community for data science and machine learning, and the BreakHis data set. Both data sets consist of labelled breast histopathology images and are described in more detail below. All images are provided in standard three channel RGB colour format. In order to test image search performance on greyscale images we use the `rgb2gray()` function in Matlab which converts RGB images to greyscale by extracting just the luminance values.

Invasive Ductal Carcinoma (IDC) Kaggle Data

The Invasive Ductal Carcinoma (IDC) dataset consists of digitized breast cancer slides from 162 patients diagnosed with IDC at the University of Pennsylvania Hospital and the Cancer Institute of New Jersey [5]. Each slide was digitized at 40× magnification, then downsampled to a resolution of 4 μm/pixel. The data set provides whole slide images split into patches of size 50 pixels by 50 pixels in RGB colour space. The supplied data was randomly split into three different subsets of 84 patients for training, 29 for validation and 49 test cases for final evaluation. Ground truth annotation regarding the presence of invasive ductal carcinoma (IDC) in each patch was obtained by manual delineation of cancer regions performed by expert pathologists. Figure 4.1 shows some sample images from the IDC data set. The images in the top row depict healthy tissue or non-invasive tumour tissue, while the images in the bottom row depict samples of invasive tumour tissue (i.e. IDC).

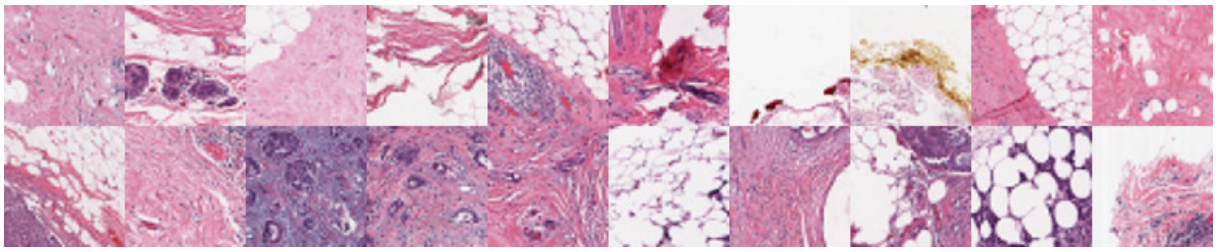

Figure 4.1: A selection of sample image patches from the IDC data set. The top row shows negative examples (healthy tissue or non-invasive tumour tissue) and the bottom row shows positive examples (IDC tissue).

Due to their small size, each individual image patch in the IDC data set may not contain both hematoxylin and eosin stains. Since the stain separation algorithm learns

the stain colours from the data, both stains must be present in the image for accurate results. To ensure good performance on all image patches, we use the entire WSI to perform stain separation and then split the image back into the original patches to compute image descriptors. One further issue is that the stain separation algorithm used assumes that two (and only two) stain components (H&E in our case) exist in the image. However, some images were observed to have significant discolouration, such as large dark patches, and the introduction of other colours not caused by H&E staining. The prevalence of such artefacts negatively impacts the ability of the stain separation algorithm to provide good results for some patients, so they were removed by searching for images which have minimal variation across the RGB channels (i.e. large black or white regions). A total of 686 patches were flagged and removed from the total data set, all of which were found to contain significant artefacts or discoloration.

Breast Cancer Histopathology Database (Breakhis):

The Breast Cancer Histopathology Database (BreakHis) [43] was built as a collaboration between researchers at the Federal University of Parana (UFPR) and the P&D Laboratory - Pathological Anatomy and Cytopathology, in Parana, Brazil. To date, it contains 9,109 images of breast tumour tissue from 82 patients using four different magnification factors: 40×, 100×, 200×, and 400×. The images are provided in PNG format (3-channel RGB, 8-bit depth/channel) and are 700×460 pixels. The data is divided into two classes, benign tumours and malignant tumours, with class labels provided by pathologists from the P&D Laboratory. Within each class, further labelling is provided to indicate tumour types. The data set consists of four histologically distinct benign tumours and four malignant tumour types. These additional intra-class labels are not used in the current study. Figure 4.2 shows some sample image patches from the BreakHis data set. The top two rows depict benign tumours, while the images in the bottom two rows show malignant tumours.

For our purposes, we will only use the subset of images taken at 40× magnification. This subset of the data contains 1,995 images, of which 652 are benign and 1,370 are malignant. Using methods provided by the authors of [43], the data has been split into a training set (consisting of 70% of the data) and testing set (consisting of 30% of the data), with the condition that there is no overlap of patients between the training and test data. The results presented for the BreakHis data set in the following sections are the average of the results over five trials where we have used the same five data folds from [43].

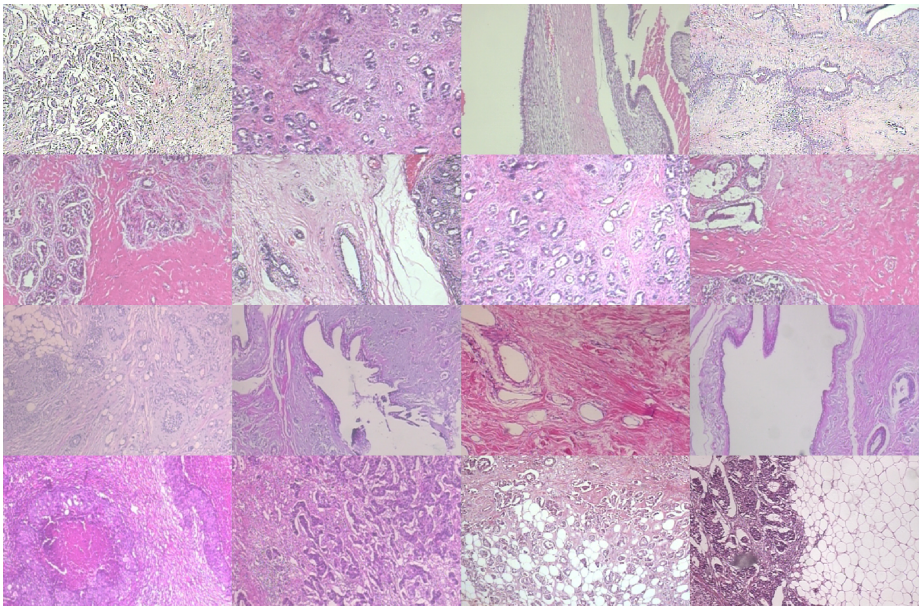

Figure 4.2: Sample image patches from the BreakHis data set. The top two rows show examples of benign tumours and the bottom two rows show malignant tumours.

4.1.2 Computation of Image Descriptors

In order to evaluate our proposed approaches we have computed the ELP and F-ELP descriptors for the images in each of the aforementioned data sets. The ELP descriptor was implemented based on Matlab code obtained from the authors of [13]. The F-ELP descriptor was implemented in a similar fashion with changes based on the description in Section 3.1.2. Based on the results for the ELP descriptor in [13] both algorithms were implemented with a window size of $n = 9$. Local windows were selected in a non-overlapping manner and a threshold of $T_H = 1$ was used to eliminate any homogeneous windows.

We note here that there is a small difference between the two algorithms in the computation of the Radon projection gradients. As was shown in Figure 3.1, when computing the Radon transform, zero-padding is added so that each projection is the same length as the longest projections (which occur along the diagonals of the image). This results in extraneous zeros lining the shorter projections which provide no useful information. To compensate for this, in our algorithm we trim the projections to the same size as the local windows, i.e. each projection has length n. As a result, the projection gradients have length $n - 1$ and the overall length of the F-ELP descriptor is $4(n - 1)$ (due to the four angles used). In the original code for the ELP algorithm, the Radon projections

are trimmed to a length of $n+1$, meaning that the corresponding projection gradients have length n and the resulting length of the descriptor is $4\,(2^{n-1})$ (again due to the four angles used). For consistency with previous works we chose not to modify the original ELP algorithm.

In addition to the ELP and F-ELP descriptors, in Section 4.3 we implement two additional image descriptors for the purposes of comparison. We choose two descriptors which have previously shown good performance on textures and patterns, the LBP descriptor and the GIST descriptor. The LBP descriptor is a histogram descriptor which describes each pixel by a binary pattern based on thresholding of the neighbouring pixels. The resulting binary patterns are binned into a histogram. The LBP descriptor is described in further detail in [20], and our implementation of the algorithm is based on this description. The GIST descriptor, previously described in Section 2.2.1, was implemented using code retrieved from the the publicly available Matlab toolbox for the LabelMe dataset [44]. All four descriptors were normalized according to the L_1 norm.

4.1.3 Evaluating Image Search Performance

Evaluating the performance of an image search algorithm is actually quite a difficult task as there is typically no ground truth information regarding the similarity of images in a data set. The ideal scenario would be to have trained experts, i.e. pathologists in this case, evaluate the results, however that is far too time-consuming and costly to be practical. It is common, instead, to indirectly evaluate image search algorithms using classification accuracy. To replicate the image search process, each query image is classified based on the mode of the class labels of its closest neighbours (where the definition of closest neighbours is dependent on the image descriptors and distance function used). If the classification accuracy is high, then it is likely that the search algorithm is performing well. We note that this approach to evaluating image search performance is more reliable when using data sets which have highly specific labels (i.e. low intra-class variation).

More formally, the approach described above is referred to as the k-Nearest Neighbours (kNN) algorithm [45]. The kNN algorithm involves searching through the training data partition and classifying each query image based on the class of its k nearest neighbours. In this work we implement the kNN algorithm using the built-in `knnsearch()` function in Matlab which allows the distance function for search to be defined as an input. Each distance function is implemented based on the definitions in Section 2.2.2.

4.1.4 Accuracy Calculations

For consistency with previous works, we use different accuracy measures to evaluate the success of the kNN algorithm for each data set.

IDC

For the IDC data, we use both the balanced accuracy (BAC) and F-measure (F1) to evaluate classification accuracy [5]. The balanced accuracy is defined as

$$\text{BAC} = \frac{\text{Sen} + \text{Spc}}{2}, \tag{4.1}$$

where Sen is the sensitivity, or true positive rate,

$$\text{Sen} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}, \tag{4.2}$$

and Spc is the specificity, or true negative rate,

$$\text{Spc} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false positives}}. \tag{4.3}$$

The F-measure, or $F1$ score is the harmonic mean of precision and sensitivity. It is often written as

$$F1 = \frac{2 \cdot \text{Pr} \cdot \text{Rc}}{\text{Pr} + \text{Rc}}, \tag{4.4}$$

where Pr is the precision,

$$\text{Pr} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false positives}}, \tag{4.5}$$

and Rc is the recall, another name for the sensitivity, i.e. Equation (4.2).

BreakHis

For the BreakHis data, we compute patient scores and the global recognition rate, both of which were introduced in [43]. If we let N_P be the number of images of patient P and

N_{rec} be the number of images of patient P that are correctly classified, then the patient score for patient P is defined as

$$\text{Patient Score} = \frac{N_{\text{rec}}}{N_P} \tag{4.6}$$

and the global recognition rate (GRR) as

$$\text{Global Recognition Rate} = \frac{\sum \text{Patient scores}}{\text{Total number of patients}}. \tag{4.7}$$

In addition to the global recognition rate we also compute the balanced accuracy as was defined for the IDC data set in Equation (4.1).

4.2 Evaluating the Effectiveness of the F-ELP Descriptor

In this Section we demonstrate the effectiveness of our frequency-based approach to encoding projection gradients. We have already commented on the compactness of descriptors generated in this way, however in this section we will also show the F-ELP generally provides better image search performance when compared with the ELP descriptor. To generate these results, the kNN image search algorithm was tested using multiple values of k. As expected, search performance tends to improve as k is increased, up to a point. There is, of course, a balance to strike between including sufficient images to accurately classify the query image, but not including images which are too dissimilar which may skew the classification results. We found that setting $k = 15$ provided generally good results. Therefore, in this and the following sections we present our findings using $k = 15$.

IDC

First, we present our results on the IDC data set. In Table 4.1 we show the resulting accuracy measures ($F1$ and BAC) for each descriptor. The best $F1$ score and balanced accuracy for each distance function, i.e. in each row, has been highlighted in bold. In Table 4.1 the descriptors have been computed for greyscale versions of the images from the data set. We can see that no matter which distance function is used for image search, the F-ELP descriptor consistently outperforms the ELP descriptor, often by quite a significant amount.

Table 4.1: $F1$ & BAC results for kNN search ($k = 15$) using the ELP and F-ELP descriptors computed for greyscale images from the IDC data set. The highest $F1$ and BAC scores in each row are indicated with bold text.

Distance Function	ELP		F-ELP	
	$F1$	BAC	$F1$	BAC
L_1	0.1250	0.5304	**0.4104**	**0.6156**
L_2	0.2954	0.5788	**0.4056**	**0.6139**
Chi-Squared	0.1001	0.5243	**0.4180**	**0.6192**
Cosine	0.1165	0.5282	**0.4092**	**0.6151**
Correlation	0.1081	0.5262	**0.4218**	**0.6199**
Hutchinson	0.3839	0.6069	**0.3976**	**0.6100**

In Tables 4.2 and 4.3 we present similar results for the H&E stain separated images and the original RGB colour images. In both cases we once again see that the F-ELP descriptor, despite being significantly shorter in length, has higher discrimination power than the ELP descriptor.

Table 4.2: $F1$ & BAC results for kNN search ($k = 15$) using the ELP and F-ELP descriptors computed for H&E stain separated images from the IDC dataset. The highest $F1$ and BAC scores in each row are indicated with bold text.

Distance Function	ELP		F-ELP	
	$F1$	BAC	$F1$	BAC
L_1	0.1054	0.5257	**0.6609**	**0.7601**
L_2	0.4194	0.6263	**0.6598**	**0.7593**
Chi-Squared	0.0599	0.5141	**0.6697**	**0.7665**
Cosine	0.0846	0.5203	**0.6637**	**0.7623**
Correlation	0.0688	0.5164	**0.6660**	**0.7641**
Hutchinson	0.4943	0.6589	**0.6593**	**0.7591**

Table 4.3: $F1$ & BAC results for kNN search ($k = 15$) using the ELP and F-ELP descriptors computed for RGB images from the IDC dataset. The highest $F1$ and BAC scores in each row are indicated with bold text.

Distance Function	ELP		F-ELP	
	$F1$	BAC	$F1$	BAC
L_1	0.1445	0.5364	**0.6369**	**0.7441**
L_2	0.4280	0.6304	**0.6338**	**0.7421**
Chi-Squared	0.0946	0.5233	**0.6343**	**0.7424**
Cosine	0.1187	0.5295	**0.6395**	**0.7458**
Correlation	0.0984	0.5244	**0.6404**	**0.7462**
Hutchinson	0.4912	0.6569	**0.6223**	**0.7339**

BreakHis

In Tables 4.4, 4.5 and 4.6 we present similar results for the BreakHis data set. Again, the best results for each distance function are highlighted in bold. We note that the balanced accuracies are generally a bit lower on this data set compared to the previous results for the IDC data set. This is to be expected due to the nature of the BreakHis data. The two classes we are attempting to distinguish between are benign and malignant tumours, which we expect to be more similar - visually - than the two classes of healthy and tumour tissue present in the IDC data set.

Table 4.4: GRR & BAC results for kNN search ($k = 15$) using the ELP and F-ELP descriptors computed for greyscale images from the BreakHis data set. The highest GRR and BAC scores in each row are indicated with bold text.

Distance Function	ELP		F-ELP	
	GRR	BAC	GRR	BAC
L_1	0.6893	**0.5747**	**0.6955**	0.5584
L_2	0.6888	**0.5737**	**0.6902**	0.5528
Chi-Squared	0.6947	**0.5797**	**0.6954**	0.5616
Cosine	0.6847	**0.5700**	**0.6948**	0.5559
Correlation	0.6867	**0.5704**	**0.6882**	0.5499
Hutchinson	**0.7085**	**0.5898**	0.6957	0.5568

On the BreakHis data set we observe that the F-ELP descriptor either performs very similarly to, or outperforms, the ELP descriptor. In Tables 4.5 and 4.6 we see that when more than one colour channel is used (i.e. when more diagnostically relevant information is

included) the F-ELP descriptor provides a higher classification accuracy the majority of the time. These results, once again, indicate that the frequency based approach used in the F-ELP algorithm is a better approach than the standard MinMax approach used in the ELP algorithm, both due to the resulting shorter histograms, and generally superior performance.

Table 4.5: GRR & BAC results for kNN search ($k = 15$) using the ELP and F-ELP descriptors computed for H&E stain separated images from the BreakHis data set. The highest GRR and BAC scores in each row are indicated with bold text.

Distance Function	ELP		F-ELP	
	GRR	BAC	GRR	BAC
L_1	0.7618	**0.7008**	**0.7670**	0.6978
L_2	0.7634	0.7006	**0.7721**	**0.7031**
Chi-Squared	0.7660	**0.7060**	**0.7737**	0.7033
Cosine	0.7618	0.7020	**0.7718**	**0.7026**
Correlation	0.7612	0.7011	**0.7706**	**0.7025**
Hutchinson	0.7387	0.6585	**0.7631**	**0.6853**

Table 4.6: GRR & BAC results for kNN search ($k = 15$) using the ELP and F-ELP descriptors computed for RGB images from the BreakHis data set. The highest GRR and BAC scores in each row are indicated with bold text.

Distance Function	ELP		F-ELP	
	GRR	BAC	GRR	BAC
L_1	0.6838	0.5901	**0.7662**	**0.6941**
L_2	0.6780	0.5839	**0.7723**	**0.6980**
Chi-Squared	0.6892	0.6008	**0.7679**	**0.6969**
Cosine	0.6873	0.5881	**0.7749**	**0.7005**
Correlation	0.6871	0.5884	**0.7740**	**0.6988**
Hutchinson	0.6873	0.5865	**0.7701**	**0.7033**

At this point it would be prudent to point out two additional observations which can be made from the above results. Both of these observations will be investigated in further detail in the following sections of this thesis:

1. We notice that image search performance is significantly improved by the use of the H&E stain separated images as compared to greyscale images, and is generally

comparable to using the original RBG images despite requiring only two colouring channels (see Section 4.3).

2. The ELP descriptor appears to be much more sensitive to the choice of distance function used in the image search algorithm than the F-ELP descriptor (see Section 4.4).

4.3 The Impact of Digital Stain Separation on Image Search Performance

The main goal of the second part of this study is to investigate the effectiveness of using stain separated images in isolation from the other parameters of the image search process. To this end, we implement our proposed F-ELP descriptor and its predecessor, the ELP descriptor, along with two additional descriptors which are known to perform well on images of textures and patterns, the LBP descriptor [20] and the GIST descriptor [22]. We recall that histopathology images, like those shown in Figure 2.1, are more like textures than typical natural images. We compare the effects of using H&E stain separated images as inputs for all four of these detectors. Table 4.7 lists each of the descriptors used and their respective lengths on each set of input colour channels that we test. We have chosen a set of descriptors with a wide variety of lengths, ranging from the very short LBP descriptor to the much longer ELP histograms.

Table 4.7: A list of the image descriptors used in this study and the corresponding number of features computed (i.e. the length of the feature vector).

Descriptor	Number of Features		
	Greyscale	H&E Stains	RGB Image
ELP	1024	2048	3072
GIST	512	1024	1536
F-ELP	32	64	96
LBP	18	36	54

We note that the length of each image descriptor is dependent on certain parameters of the algorithm. In this work the following parameters are used: the ELP and F-ELP descriptors are implemented, as before, with a window size of $n = 9$, the GIST descriptor, by default, divides the image into a 4×4 grid and uses a filter bank of 32 Gabor filters, and the LBP descriptor is computed using $P = 16$ neighbouring pixels at a radius of $R = 2$ pixels away from the centre pixel. We notice in Table 4.7 that as we increase the number of

input colour channels from one to three, the length of the feature vectors increases. Given that the computation time for the kNN search algorithm has linear dependency on feature vector length [46], it is clear that for a fixed image descriptor, it is desirable to use fewer colour channels, so long as this does not cause the overall search performance does not suffer significantly.

IDC

In Table 4.8 we present the best global recognition rates and balanced accuracies for each image descriptor and set of input colour channels for the IDC data set. For each descriptor, and each set of input colour channels, the best accuracy, taken over all distance functions, is presented. The highest $F1$ score and balanced accuracy for each descriptor, i.e. in each column, is highlighted in bold. As expected, since coloured images contain relevant information which is lost when images are converted to greyscale, we observe that using either the H&E stain separated images or the total RGB images is always an improvement over using greyscale images.

Table 4.8: The best KNN search ($k = 15$) accuracy for the IDC dataset taken over all distance functions. The top result in each column is highlighted in bold.

Colour Channels	ELP		GIST		F-ELP		LBP	
	$F1$	BAC	$F1$	BAC	$F1$	BAC	$F1$	BAC
Greyscale	0.3839	0.6069	0.5910	0.7207	0.4218	0.6199	0.5396	0.6825
H&E Stains	**0.4943**	**0.6589**	**0.6283**	**0.7570**	**0.6697**	**0.7665**	0.6887	0.7780
RGB Image	0.4912	0.6569	0.6147	0.7448	0.6404	0.7462	**0.7125**	**0.7972**

A more interesting comparison comes from looking at the bottom two rows of the table, comparing the use of the H&E images to the RGB images. We see that generally the $F1$ scores and balanced accuracies are similar (within one or two percentage points) for both the H&E stain separated and RGB images. In most cases, we actually observe an improved performance using the H&E image over RGB, despite the fact that the input image has fewer colour channels, and thus the feature vector is shorter. Clearly using the stain separated images is the superior choice, as the descriptors are more compact, and performance is similar, if not improved compared to using the original images.

BreakHis

Similarly, Table 4.9 presents the results comparing the use of greyscale, stain separated and RGB images as inputs to the image search algorithm for the BreakHis data set. As before, we note that the balanced accuracies are lower than those for the IDC data set.

Table 4.9: The best KNN search ($k = 15$) accuracy for the BreakHis dataset taken over all distance functions. The top result in each column is highlighted in bold.

Colour Channels	ELP		GIST		F-ELP		LBP	
	GRR	BAC	GRR	BAC	GRR	BAC	GRR	BAC
Greyscale	0.7085	0.5898	0.7128	0.5744	0.6957	0.5616	0.7051	0.5979
H&E Stains	**0.7660**	**0.7060**	**0.7406**	**0.6286**	0.7737	**0.7033**	**0.7564**	**0.6885**
RGB Image	0.6892	0.6008	0.7068	0.6090	**0.7749**	**0.7033**	0.7023	0.6219

Once again, we observe a general increase in image search performance when more than one input colour channel (H&E or RGB) is used as compared to the use of single channel greyscale images as inputs. On this data set we see that the use of the stain separated images often leads to a significant increase in classification accuracy over the RGB images. It is likely that this observation is due to the fact that there is more variation in the appearance of the H&E stain colours across the BreakHis data set than in the IDC data set. This difference is apparent when we compare the images in Figure 4.1 and 4.2. As expected, the benefits of using H&E stain-separation become more apparent when there are significant variations in the stain colours, as we would expect to see when implementing image search in practice on data from multiple sources.

4.4 Comparison of Distance Functions for Image Search

Previously, we saw that the performance accuracy of the ELP descriptor was notably affected by the choice of distance function used, whereas we did not see the same for the F-ELP descriptor. In this section, we investigate this further by looking at how the choice of distance function affects the results for all four descriptors tested. To do so, we introduce a ranking of each distance function based on how it performs relative to the other distance functions. For each search trial, i.e., each image descriptor and each selection of input colour channels (greyscale, H&E, or RGB), we rank the distance functions based on their

accuracy as a percentage of the maximum accuracy over all distance functions for that particular search trial. We found that similar results and trends were observed regardless of the input colour channels used, therefore in the following we present an average distance ranking, which is the average of the distance ranking taken over the three possible choices of input images.

As the balanced accuracy was computed for both the IDC and BreakHis data sets, we present an average distance ranking based on the balanced accuracy resulting from each search trial. We note that although we do not include them here, the results for the $F1$ measure (IDC data) and global recognition rate (BreakHis data) are found to follow similar trends.

IDC

First, we present our results on the IDC data set. Figure 4.3 shows the average ranking of all

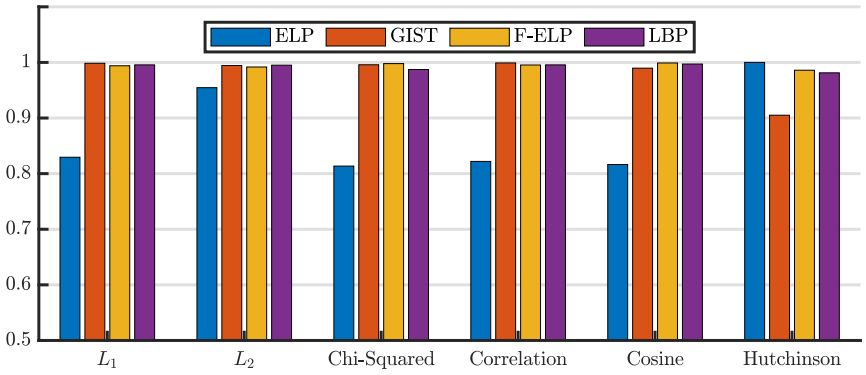

Figure 4.3: A comparison of the average BAC ranking of distance functions for each image descriptor on the IDC data set.

six distance functions for each of the ELP, GIST, F-ELP and LBP descriptors. Somewhat surprisingly, we observe that, in general, the variation in search accuracy across distance functions is relatively low. It would be difficult to pinpoint any one distance function which is better than the others. We do notice that, as we saw in Section 4.2 the ELP descriptor appears to be the most sensitive to the choice of distance function, performing best when the Hutchinson distance is used. For the other three descriptors, we see that the variation in performance based on the choice of distance function is almost non-existent.

BreakHis

In Figure 4.4 we show the average rankings of each distance function on the BreakHis data. Once again, the overall change in the accuracy as a result of the choice of distance

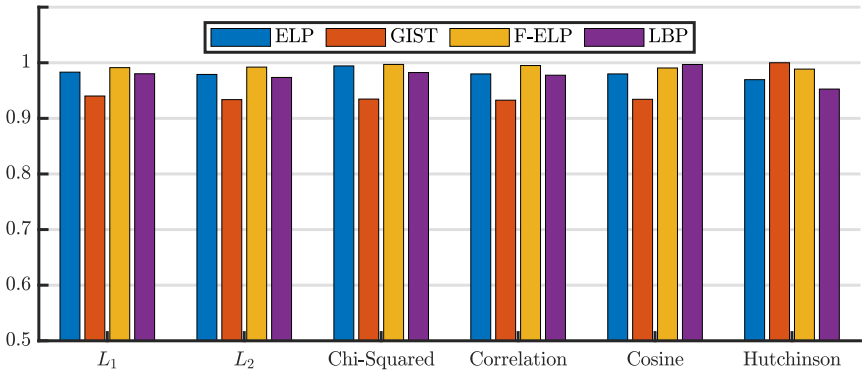

Figure 4.4: A comparison of the average ranking of distance functions for each image descriptor on the BreakHis data set.

function is surprisingly low for all descriptors. On this data set, we we see that the ELP descriptor performs similarly regardless of distance function, while the performance of the GIST descriptor is slightly more variable. In all cases, the impact of the distance function on search performance is small, and generally inconsistent across both descriptors and data sets.

Our results comparing the use of various distance function for image search on both the IDC and BreakHis data sets do not give any indication that one distance function is necessarily superior for image search - even for a fixed choice of image descriptor. Over many tests, we see only one scenario (the ELP descriptor applied to the IDC data) where the choice of distance function significantly impacts the results. In the future, subjective testing by experts of image search results may reveal more significant differences. However, until that is possible, our results indicate that in practical applications where the algorithm must generalize to data from multiple sources, the best choice is simply to use the distance function which can be computed most efficiently.

4.5 Conclusions and Applications to Future Research

In Part I of this thesis we have carried out a critical examination of the design of an image search algorithm for digital histopathology. Specifically, we have investigated the design of

image descriptors, the use of digital stain separation algorithms to separate images into physically meaningful colour components, and looked at how the choice of distance function may impact the effectiveness of an image search algorithm. To this end, we propose an image descriptor using a frequency-based encoding of local projection data, the F-ELP descriptor. At the time of this work, the F-ELP descriptor was a significant improvement over its predecessor, the ELP descriptor [13] and comparable to the state-of-the-art in the field, including some deep learning algorithms. Although there have been advancements in the field since then, some of the results of our study are still very applicable to future developments.

Firstly, we have shown that our proposed frequency-based encoding of projection data is more efficient and more effective than the standard MinMax binary encoding used to generate histogram descriptors. We once again mention that due to the lack of large, balanced data sets with detailed annotations and the need for "explainability" in the medical field, there is still a need for handcrafted descriptors in the digital histopathology space. For use in large scale medical applications these descriptors should be compact, explainable, and generalizable to data from multiple sources. Our frequency-based method of encoding transform data can be extended to efficiently encode any sort of projection or image transform data, not just the Radon transform from the ELP method. Additionally, the information from this method of encoding is more physically relevant than a binary encoding, and therefore more explainable when it comes to justifying a diagnostic decision.

Arguably the most important contribution of this work is the result in Section 4.3 on using digitally stain separated images for image retrieval applications. After testing on both the IDC and BreakHis data sets, we consistently find that separating images into their H&E stain components leads to a significant increase in search performance over simply using the greyscale images, as expected. More interestingly, we find that using H&E separated images yields search accuracies within one or two percent of those obtained with the original RGB images, despite the fact that the H&E images have only two colour channels. In fact, superior performance is observed using the H&E images in most tested scenarios. Given the improved computation speed afforded by using fewer image channels, it is reasonable to conclude that using H&E stain separated images is preferable to using the overall RGB images for image search. We expect that these results will generalize to additional descriptors and approaches for the retrieval and classification of digital histopathology images, even deep learning. In fact, in the years since the publication of our results [2] others have used stain separation of digital histopathology images with good results. For instance, in [39] the authors use stain separation in order to train CNNs which can generalize to data with colour variations and in [47] stain separation is used prior to computing texture features for the classification of breast cancer histopathological

images.

Finally, we have tested our image search algorithm using a variety of distance functions, from the simple L_1 and L_2 distance functions, to the more complex Hutchinson distance. Somewhat surprisingly, we found that the choice of distance function did not have a significant or consistent impact on the image search performance when tested using multiple image descriptors and on two histopathology data sets. From this we conclude that in applications where the goal is good performance on large data sets consisting of data from multiple sources, the best choice is to use whichever distance function can be computed the most efficiently.

Part II

Estimating the Fractal Dimensions of Vascular Networks and Other Branching Structures

Chapter 5

The Basics of Fractal Geometry

Ever since Benoît Mandelbrot first coined the term fractal (from the Latin term *fractus*, meaning broken) in 1975 [48], mathematicians have struggled to reach a consensus as to the formal definition of a fractal. In fact, Mandelbrot himself is purported to have proposed multiple definitions. In his book *The Fractal Geometry of Nature* he states

> *A fractal is by definition a set for which the Hausdorff-Besicovitch dimension strictly exceeds the topological dimension.* [49] (p. 15),

yet only a few years later he is said to have retracted this definition, citing it as too restrictive, and proposed the following instead

> *A fractal is a shape made of parts similar to the whole in some way.* [50] (p. 11).

Even now, many years later, it would seem that we are no closer to having a single, precise definition for what makes something a fractal. As Kenneth Falconer writes in the introduction to his book, *Fractal Geometry, 2014*:

> *... the definition of a 'fractal' should be regarded in the same way as a biologist regards the definition of "life". There is no hard-and-fast definition but just a list of properties characteristic of a living thing...* [51] (p. xxvii).

This then begs the question, what are these properties which are characteristic of fractals?

5.1 Fractal Sets and Fractal Properties

Although the precise definition of a fractal is not clear - ask any mathematician, and surely they will be able to provide you with a list of many well known fractal sets. Therefore, in order to get an understanding of what is meant by the term fractal, we will begin by looking at some common examples of fractal sets and making note of their properties. We will see that these fractal sets are generally too irregular to be adequately described using classical geometry, and are understood using the framework of fractal geometry.

The middle third Cantor set, often referred to as just the Cantor set, is one of the most well known examples of a fractal set. It is easiest to explain the Cantor set by describing its construction. We start with the unit interval $[0, 1]$, which we denote by E_0, and construct a new set, E_1, by removing the middle third of E_0. E_1 therefore consists of the two intervals $\left[0, \frac{1}{3}\right]$ and $\left[\frac{2}{3}, 1\right]$. Repeating the process, E_2 is obtained by removing the middle thirds of these intervals so that E_2 consists of the four intervals $\left[0, \frac{1}{9}\right]$, $\left[\frac{2}{9}, \frac{1}{3}\right]$, $\left[\frac{2}{3}, \frac{7}{9}\right]$ and $\left[\frac{8}{9}, 1\right]$. We continue in this way, constructing the set E_k by the removal of the middle third of each interval in E_{k-1} such that each set E_k consists of 2^k intervals of length 3^{-k}. The Cantor set is the limit of the sequence of sets E_k, shown in Figure 5.1, as k tends to infinity. It can be shown that the Cantor set is an infinite (and uncountable) set [51]. We can see that the

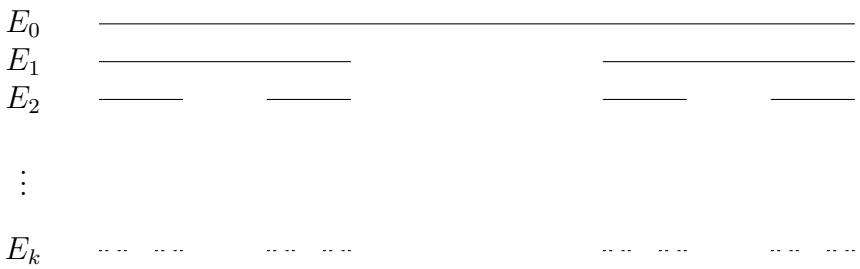

Figure 5.1: The process of constructing the middle third Cantor set.

resulting set is highly irregular and difficult to describe in classical terms. Here, we make note of some interesting properties of the middle third Cantor set which we will see are common properties possessed by most fractal sets.

1. The Cantor set is comprised of scaled copies of itself. For example, the parts of the set in the intervals $\left[0, \frac{1}{3}\right]$ and $\left[\frac{2}{3}, 1\right]$ are each copies of entire set scaled by a factor of $\frac{1}{3}$. This property is called self-similarity.
2. The Cantor set has a "fine structure", i.e. as we zoom in on the set more detail emerges.

3. The Cantor set contains an uncountably infinite number of points, yet the length of the set is zero. The size of the Cantor set is difficult to quantify using classical metrics (such as length).

Another example of a well known fractal set is the von Koch Curve. Again, we start with the unit interval, $E_0 = [0, 1]$, and construct the set in a recursive manner as shown in Figure 5.2. On the first iteration E_1 is constructed by replacing the middle third of E_0

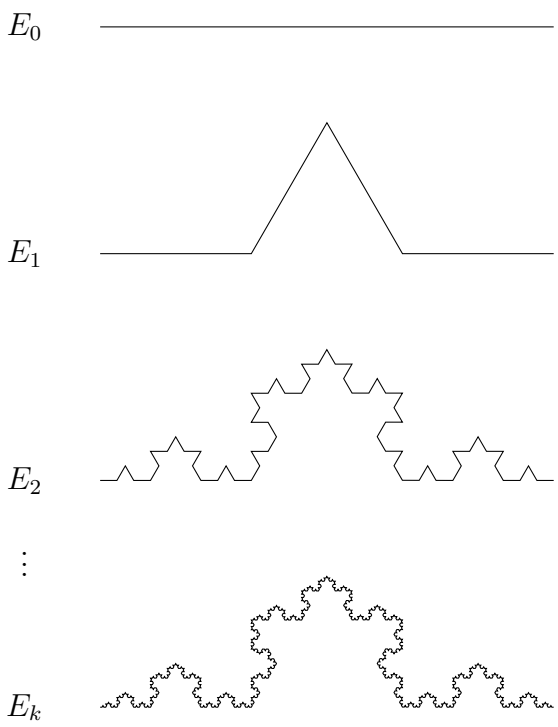

Figure 5.2: Construction of the von Koch Curve.

with two line segments which would form an equilateral triangle with the removed line segment. We repeat this process, replacing each line segment with the other two sides of an equilateral triangle to construct the set E_k from E_{k-1}. If we let k approach infinity, then this sequence of polygonal curves E_k approaches a limiting curve, which is called the von Koch curve. The von Koch curve has many similarities with the middle third Cantor set described above. We can see from Figure 5.2 that the limiting set will be comprised of four copies of itself, each of which has been scaled by a factor of $\frac{1}{3}$. Due to the recursive nature of its construction, the von Koch curve possesses a fine structure - more detail emerges as

we zoom in. In fact, we can show that each iteration E_k has length given by

$$L_k = \left(\frac{4}{3}\right)^k, \tag{5.1}$$

and as we let k tend to infinity, the length of the limiting curve, i.e. the von Koch curve is

$$L = \lim_{k\to\infty} L_k = \infty. \tag{5.2}$$

The von Koch curve has infinite length - in other words the von Koch curve is a non-rectifiable curve.

On the other hand, if we estimate the area of the von Koch curve by covering the set with square tiles of size $\varepsilon_k \times \varepsilon_k$, the area can be approximated as

$$A_k = N(\varepsilon_k)\varepsilon_k^2 \tag{5.3}$$

where $N(\varepsilon_k)$ is the number of tiles of size ε_k required to cover the set. If we let $\varepsilon_k = 1/3^k$ then $N(\varepsilon_k) = 4^k$ and

$$A_k = \left(\frac{4}{9}\right)^k. \tag{5.4}$$

The area of the set can be found by letting k go to infinity, or letting the square tiles get infinitesimally small, i.e.

$$A = \lim_{k\to\infty} A_k = 0. \tag{5.5}$$

The von Koch curve therefore has zero area, so neither its length nor area provides us with any meaningful information. These results might indicate that the von Koch curve is "thicker" than a curve (infinite length) and "thinner" than a planar area (zero area) and that a fractional ("fractal") dimension somewhere between one and two may be better suited to describe the set. We shall discuss this in more detail in Section 5.2.

Some other commonly recognized fractal sets can be constructed using similar recursive procedures. One example is the Sierpinski triangle (see Figure 5.3 (A)), which is constructed by the repeated removal of subsequently smaller inverted equilateral triangles from an initial equilateral triangle. Another example is the equivalent of the Cantor set in $\mathbb{R}^2$, referred to as a Cantor dust, shown in Figure 5.3 (B). We can see that each of these examples also embodies similar properties to the Cantor set and the von Koch curve, as described above, including an irregular geometry, self-similarity, a fine structure, and a size which is difficult

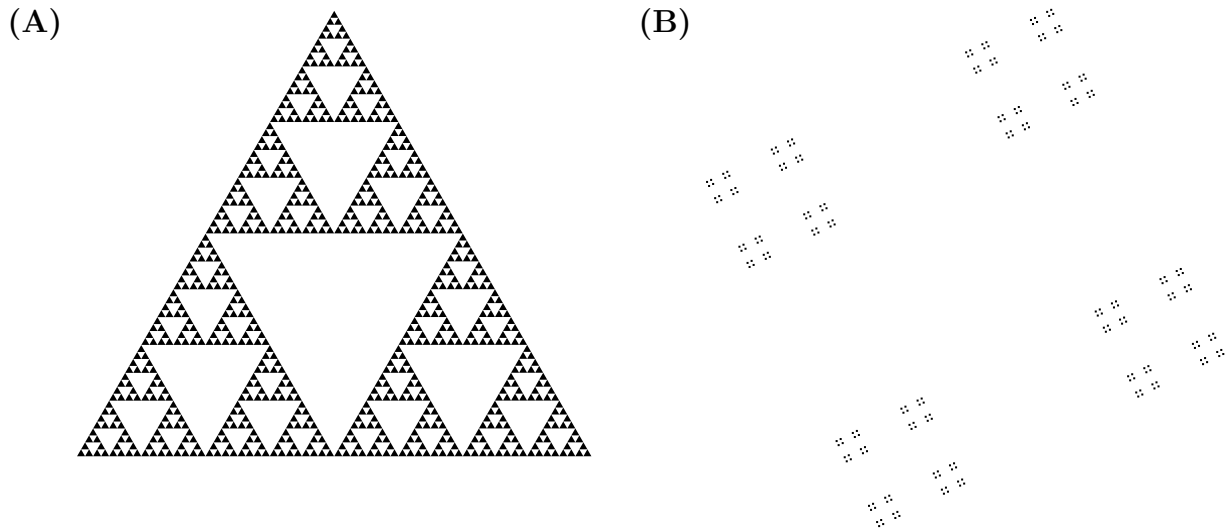

Figure 5.3: Two additional fractal sets: **(A)** the Sierpinski triangle, and **(B)** a Cantor dust.

to quantify using traditional means. These are the properties we will generally keep in mind when we refer to fractal sets, however not all fractal sets will necessarily have all of these properties exactly. For instance, although many fractals have some degree of self-similarity, in many cases the the similarity may be only approximate or statistical in nature.

We should note here that in the previous discussion the Cantor set and the von Koch curve were described as the limits of recursive procedures involving generators. A generator, G, is a set of rules pertaining to operations on sets, i.e. "remove the middle third of each line segment". This is one way of constructing fractal sets, however a more common method of generating (and analysing) fractal sets is through the application of iterated function systems (IFS). Quite simply, an IFS is a system of contraction mappings which act on a set in a parallel manner. The term iterated function system comes from M. Barnsley and S. Demko [52], however the idea was first developed in an earlier paper by J. Hutchinson [53]. Although the study and application of iterated function systems is an interesting subject in its own right, the generation of fractal sets is not the focus of this thesis, and thus these ideas will not be discussed in further detail.

5.1.1 Self-Similarity and Statistical Self-Similarity

The notion of self-similarity will be of particular importance throughout the remainder of this thesis, so we will spend a bit more time here describing the ways in which a set may be

self-similar. In general, when we refer to a set as being self-similar or having some degree of self-similarity, this means that it is comprised of parts which resemble (are similar to) the whole in some way. Strict geometrical self-similarity, or just strict self-similarity, refers to sets like the Cantor Set and von Koch curve in which the parts are exact copies of the whole, transformed only by a scaling and/or rotation operation. Specifically,

> *A [set] is strictly self-similar if the [set] can be decomposed into parts which are exact replicas of the whole. Any arbitrary part contains an exact replica of the whole [set].* [54] (p. 21),

Strict self-similarity, like that described above, is a useful mathematical construct, but highly unlikely to exist outside of specific man-made examples. More realistically, we will have sets which are self-similar in an approximate or statistical sense. For example, Figure 5.4 shows a random version of the von Koch curve. This set was constructed by tossing a coin at each step in the generation process to determine whether or not the next set of lines should point inward or outward. The strict self-similarity of the von Koch curve in Figure 5.2 has been replaced with a statistical self-similarity - which means that if we zoom in on small pieces of the curve they will have the same statistical distribution as the whole set.

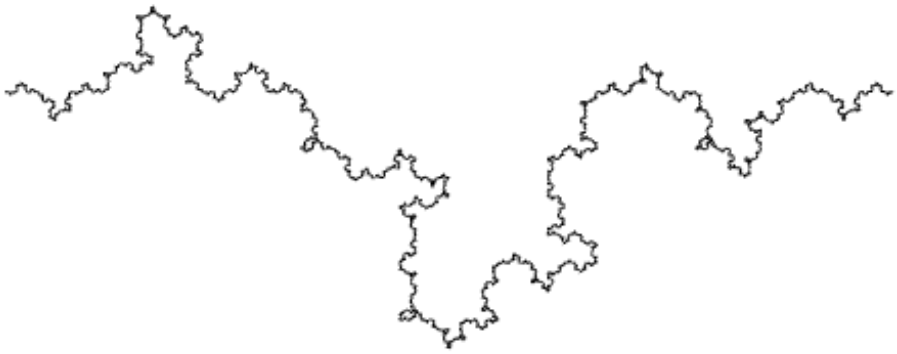

Figure 5.4: A random von Koch curve.

It is necessary to make an important distinction here between true self-similarity, which arises from the type of infinite recursive procedure described in the previous section, and what we will call "finite self-similarity". In practice, we can only work with finite approximations

of self-similar sets, where the self-similarity (whether it be strict, approximate, or statistical) only holds down to some finite resolution (or scale). For instance, the sets E_k shown in Figures 5.1 and 5.2 would be considered finitely self-similar as they appear self-similar when examined with a finite resolution. Similarly, we can say the random von Koch curve shown in Figure 5.4 is statistically self-similar down to the scale of the smallest line segment visible in the figure.

5.1.2 Fractal Trees

Although many fractals do have all of the properties described previously, there are exceptions which are still widely considered as fractals. Fractal trees are one such exception. Fractal trees are of particular interest in many areas of research as tree-like structures occur frequently in nature. For example, we observe branching in trees, rivers/streams, the vascular system, branching airways in the lungs, etc... Due to their resemblance to fractal trees, naturally occurring branching structures are often thought to have fractal properties and, as we will see, many attempts have been made to characterize such structures using fractal geometry.

A fractal tree is a set resulting from repeated application of a branching generator, like the one shown in Figure 5.5, to the branches generated in the previous step. We can see

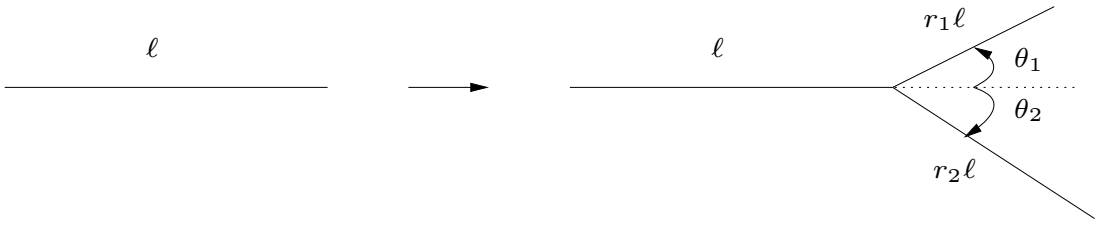

Figure 5.5: A simple branching generator with two branches ($N = 2$), branching angles θ_1 and θ_2, and contraction factors r_1 and r_2.

from Figure 5.6 that if we repeat the branching process indefinitely, each sub-tree emanating off the main trunk will be a scaled and rotated copy of the tree itself. The tree resulting from this process is referred to as an infinite binary (binary because there are two branches generated on each iteration) fractal tree. A large number of distinct fractal trees can be produced by varying the parameters of the branching generator, including the number of branches created on each iteration (N), the length scaling ratios (r_1, r_2, etc...) and the branching angles (θ_1, θ_2, etc...).

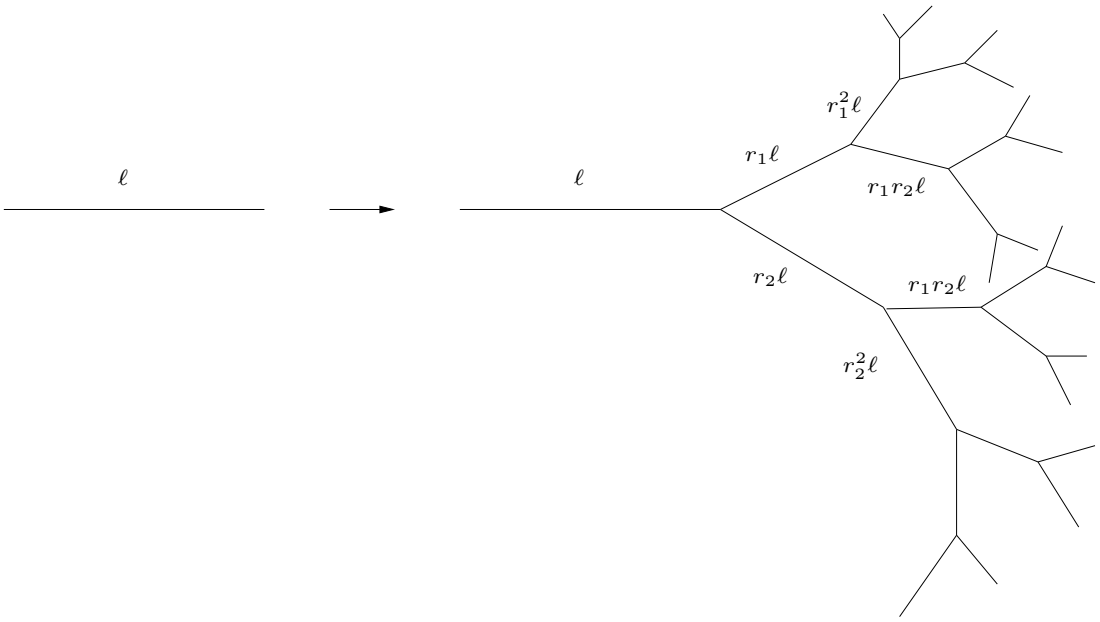

Figure 5.6: A branching structure which results from four applications of the branching generator shown in Figure 5.5.

Although they are commonly referred to as fractals, fractal trees differ from the examples of fractal sets we have seen so far. Even in the infinite case, fractal trees are not strictly self-similar (i.e. made up of a union of scaled/rotated copies of themselves) due to the fact that the trunk remains unchanged during the generation process. At best, fractal trees can be viewed as the union of two scaled copies of themselves plus a trunk, which can be thought of as a degenerate copy of the set, squashed infinitely thin in one direction. In fact, the generator in Figure 5.5 can also be viewed as a special type of IFS - a two-map IFS with condensation [55], where the condensation set, the trunk in this case, is a part of the set which remains unchanged throughout the generation process. As a result of this property, i.e. the presence of the unchanged trunk, fractal trees are sometimes referred to as "non-uniform" fractals or fractals "with residue" [49]. This property will be important later in the thesis in our discussion on the fractal dimensions of trees and other branching structures.

5.2 Fractal Dimensions

Dimension is, very roughly, a measure of how much space a set occupies near each of its points [51]. The idea of assigning dimensions to sets can be traced back as far as approximately 300 B.C.E. to the first recorded definition of points, surfaces and solids in Euclid's Elements [56]. Over the years this early definition of dimension has been updated and refined to be more precise and to encompass a wider variety of objects and topological spaces. One of the most common definitions of dimension used today is the topological dimension (sometimes also known as the Lebesgue covering dimension). The topological dimension of a set is defined based on how that set can be divided into smaller parts. In ordinary Euclidean spaces (i.e. $\mathbb{R}^1$, $\mathbb{R}^2$, $\mathbb{R}^3$, etc...) points have topological dimension zero, curves have dimension one, areas have dimension two and volumes have dimension three [57]. We note that the topological dimension of a set, by definition, is always an integer.

We have already alluded to the fact that that our usual measures of size/dimension are not particularly informative when it comes to fractal sets. For instance, both the von Koch curve and Sierpinski triangle, which have topological dimension one, can be shown to have infinite length despite occupying zero area. Neither length nor area provides us with much useful information about either of these sets. Typically, length is used to measure the size of one-dimensional sets and area is used to measure the size of two-dimensional sets, so it would seem that these two fractal sets are neither one dimensional nor two dimensional. To reconcile this phenomena with our usual definition of dimension, the use of fractional (or fractal) dimensions was proposed [58]. Fractal dimensions are one of the main tools of fractal geometry which allow for the characterization and comparison of many fractal sets. In simple terms, it is a measure of how an object fills the space it is contained in.

At this point it would be prudent to mention that there is not one singular definition of the "fractal dimension" of a set - it is possible to define the dimension of a set in many different ways and different definitions of dimension may even give different values for the same set. The term "fractal dimension" is often misused, or used inconsistently, in the literature. In fact, use of the term "fractal dimension" without further clarification should always be treated with some skepticism.

5.2.1 Similarity Dimension

For strictly self-similar fractal sets we can compute something called the similarity dimension. The similarity dimension is a measure of how the mass of a self-similar set changes as the set is scaled. If we consider a self-similar set which is the union of N non-overlapping copies

of itself such that each copy is scaled down by a factor of $r < 1$, then N and r are related by the power law $N = r^{-\dim_S}$ where $\dim_S$ is the similarity dimension.

Definition 5.2.1. *The similarity dimension of a strictly self-similar set E is*

$$dim_S(E) = \frac{-\log(N)}{\log(r)} = \frac{\log(N)}{\log\left(\frac{1}{r}\right)}. \tag{5.6}$$

The similarity dimension can be applied to any strictly self-similar set, even those which are not fractal, such as the standard Euclidean objects, lines, squares and cubes. First, let's consider a line. If we scale a line by a factor of one half, as shown in Figure 5.7, we can see that two copies are required to form the original set. Therefore, $N = 2 = (1/2)^{-1}$ and $\dim_S = 1$ according to Equation (5.6). Another way of thinking about this is that when the line segment is scaled by a factor of one half, the mass of each scaled segment is also scaled by one half, and thus we require two scaled copies to make up the whole set.

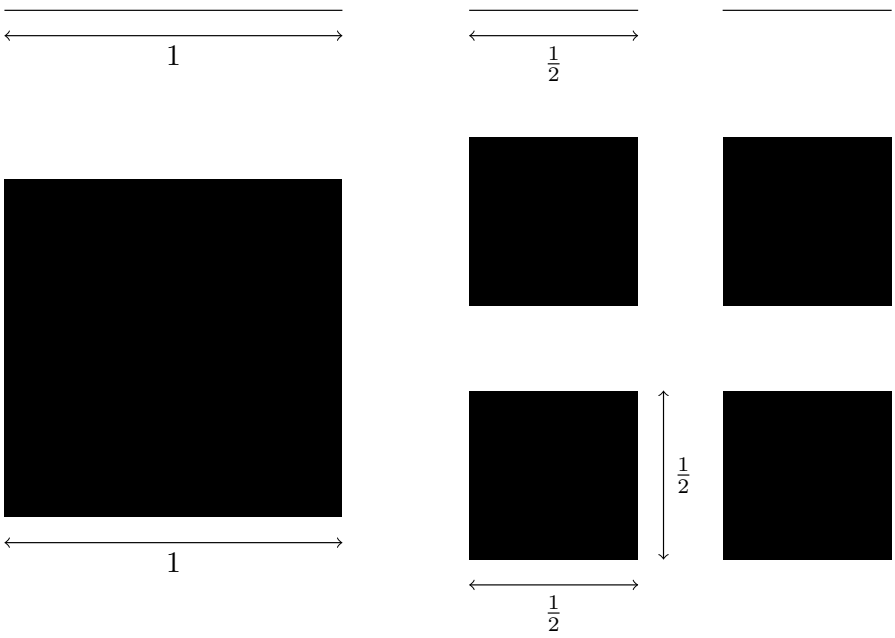

Figure 5.7: A line (a one dimensional object) can be split up into two copies of itself scaled by one half, whereas we require four copies scaled by one half to make up a square (a two dimensional object).

Now, consider the square shown in Figure 5.7. If we scale the square by a factor of one half we require four scaled copies to make up the whole set. This is because each scaled

copy of the square has one quarter (or $(1/2)^2$) the mass of the whole set. According to Equation 5.6 this gives us $\dim_S = 2$ for the square. Our results for the similarity dimension of a line and a square match up nicely with our usual notion of the dimensions of these sets. A similar exercise yields $\dim_S = 3$ for a cube, as expected.

The similarity dimension is more interesting when it is applied to fractal sets. For instance, we can see that the Cantor set from Figure 5.1 is composed of $N = 2$ copies of itself each scaled by $r = 1/3$. Therefore, the similarity dimension of the Cantor set is

$$\dim_S = \frac{-\log(2)}{\log(1/3)} = \frac{\log(2)}{\log(3)} \approx 0.631. \tag{5.7}$$

This dimension captures the idea that the Cantor set is something between a collection of points and a line. Similarly, the von Koch curve is made up of four copies of itself each scaled by a factor of $1/3$. The similarity dimension of the von Koch curve is therefore

$$\dim_S = \frac{-\log(4)}{\log(1/3)} = \frac{\log(4)}{\log(3)} \approx 1.262, \tag{5.8}$$

a dimension between one and two, as expected. As we can see, the similarity dimension allows for non-integer dimensions, providing more information about fractal sets than the standard topological dimension. We can easily compute the similarity dimension for all of the strictly self-similar fractals presented in the previous section. The Sierpinski triangle, Figure 5.3 (A), has similarity dimension $\log(3)/\log(2) \approx 1.585$ and the Cantor dust in Figure 5.3 (B) has similarity dimension $\log(4)/\log(3) \approx 1.262$, the same as the von Koch curve. It is interesting to note here that despite having the same dimension, the von Koch curve and Cantor dust are visually distinct sets and generated by very different processes. Allowing for fractional dimensions gives us more information than the topological dimension, but this does not mean that all sets with the same fractal dimension are the same (just like not all one dimensional, or two-dimensional, etc..., sets are the same).

For more general fractal sets which do not possess strict self-similarity, the similarity dimension is not applicable. In these cases, there are other definitions of dimension, such as the Hausdorff and box-counting dimensions.

5.2.2 Hausdorff Dimension

The Hausdorff dimension, also known as the Hausdorff-Besicovitch dimension, is the oldest definition of a fractal dimension [51]. It is this definition of dimension which was referenced

by Mandelbrot in his early definition of a fractal. Unlike the similarity dimension, the Hausdorff dimension is defined for any set, regardless of self-similarity or other properties. In order to define the Hausdorff dimension of a set $E \subset \mathbb{R}^n$, we will first need the following definitions.

Definition 5.2.2. *Given $U \subset \mathbb{R}^n$, the diameter of U is $|U| = \sup\{|x-y| : x, y \in U\}$. In other words, the diameter is the greatest distance between any two pairs of points in U.*

Definition 5.2.3. *If $\{U_i\}$ is a countable or finite collection of sets which cover E, i.e. $E \subset \cup_{i=1}^{\infty}$, such that $0 < |U_i| \leq \varepsilon$ for all i, then $\{U_i\}$ is an ε-cover of E.*

We will also need to define the Hausdorff measures of the set E. First we let

$$H^s_\varepsilon(E) = \inf\left\{\sum_{i=1}^{\infty} |U_i|^s : \{U_i\} \text{ is an } \varepsilon\text{-cover of } E\right\}. \tag{5.9}$$

We can see that as $\varepsilon \to 0$, the set of possible ε-covers of E is reduced, so the infimum is non-decreasing and thus approaches a limiting value. This limit is defined for any subset of $\mathbb{R}^n$ and is called the s-dimensional Hausdorff measure of the set E.

Definition 5.2.4. *The s-dimensional Hausdorff measure of E is defined as*

$$H^s(E) = \lim_{\varepsilon \to 0} H^s_\varepsilon(E). \tag{5.10}$$

The Hausdorff measures generalize the usual ideas of length, area and volume and it can be shown [51] that, for subsets of $\mathbb{R}^n$, the n-dimensional Hausdorff measure is equivalent to the n-dimensional volume (up to a multiplicative constant).

For most values of s the s-dimensional Hausdorff measure takes on the value either 0 or ∞, with one critical value of s at which $H^s(E)$ jumps from ∞ to 0. This critical value is referred to as the Hausdorff dimension of E [51].

Definition 5.2.5. *The Hausdorff dimension of the set E is the real number*

$$dim_H(E) = \inf\{s : H^s(E) = 0\} = \sup\{s : H^s(E) = \infty\}. \tag{5.11}$$

In general, the Hausdorff dimension and the similarity dimension are not equal, instead we have the following inequality

$$\dim_H(E) \leq \dim_S(E). \tag{5.12}$$

However, if E is a strictly self-similar set such that the copies (or the similarities) are disjoint, then we will have equality between the two dimensions [58].

Although the Hausdorff dimension is well defined mathematically, it is difficult to calculate, and even more difficult to estimate empirically for practical data sets which are not, in general, strictly self-similar. For this reason, use of the Hausdorff dimension is primarily limited to theoretical discussions, and other definitions of dimension are used when dealing with real-world objects. The box-counting dimension, described below, is one such dimension.

5.2.3 Box-Counting Dimension

The box-counting dimension is one of the most well known and widely used fractal dimensions due its relatively intuitive formulation and ease of empirical estimation [51] via the aptly named box-counting method (see Section 6.1). Once again, we consider E to be any non-empty subset of $\mathbb{R}^n$ and we let $N(\varepsilon)$ be the number of sets in the smallest ε-cover of E.

Definition 5.2.6. *The lower box-counting dimension of a set E is defined as*

$$\underline{dim}_B(E) = \liminf_{\varepsilon \to 0} \frac{-\log N(\varepsilon)}{\log(\varepsilon)} \tag{5.13}$$

and the upper box-counting dimension of E is defined as

$$\overline{dim}_B(E) = \limsup_{\varepsilon \to 0} \frac{-\log N(\varepsilon)}{\log(\varepsilon)}. \tag{5.14}$$

In most practical examples, the lower and upper box-counting dimensions are equal and we have the following definition for the box-counting dimension.

Definition 5.2.7. *The box-counting dimension of E is*

$$dim_B(E) = \lim_{\varepsilon \to 0} \frac{-\log N(\varepsilon)}{\log(\varepsilon)}. \tag{5.15}$$

Equation (5.15) implies that for small values of ε we have

$$N(\varepsilon) = c\varepsilon^{-s} + \mathcal{O}\left(\varepsilon^{-s}\right) \quad \text{as } \varepsilon \to 0 \tag{5.16}$$

where c is a positive constant and $s = \dim_B(E)$. In other words, the number of sets required to cover E follows, at least approximately, a power law in ε as ε gets small.

Calculating $N(\varepsilon)$ can be difficult in practice, however it turns out that there are several equivalent definitions which may be more convenient. According to [51], which includes a proof of equivalence, we can define $N(\varepsilon)$ to be any of the following and get the same result for $\dim_B(E)$:

i. The smallest number of sets of diameter at most ε which cover E.

ii. The smallest number of closed balls of radius ε which cover E.

iii. The smallest number of cubes of side length ε which cover E.

iv. the largest of number of disjoint balls of radius ε with centres in E.

v. The number of ε-mesh cubes which intersect E.

This last definition requires us to define what is meant by an ε-mesh.

Definition 5.2.8. *The family of cubes in $\mathbb{R}^n$ of the form*

$$[m_1\varepsilon, (m_1+1)\varepsilon] \times \cdots \times [m_n\varepsilon, (m_n+1)\varepsilon], \tag{5.17}$$

with $m_1, ..., m_n$ integers, is called an ε-mesh or ε-grid.

We will see that this definition of the box-counting dimension in terms of the number of ε-mesh cubes which intersect the set is quite useful when it comes to empirical estimation of the box-counting dimension.

A natural question to ask is how does the box-counting dimension relate to the Hausdorff dimension. For many regular sets, the box-counting and Hausdorff dimension are equal, however in general it can be shown [51] that the box-counting dimension is an upper bound for the Hausdorff dimension:

$$\dim_H(E) \leq \underline{\dim}_B(E) \leq \overline{\dim}_B(E) \tag{5.18}$$

with many examples where there is strict inequality. For example, the box-counting dimension of the set $\{x_n : x_n = 1/n, n = 1, 2, 3, ...\}$ is $1/2$ [57]. On the other hand the Hausdorff dimension of this set is zero, as the Hausdorff dimension of any countable set is zero [51].

5.2.4 The Spectrum of Generalized Fractal Dimensions

All of the fractal dimensions we have discussed so far have involved just a single scaling exponent, however not all objects can be described with just one exponent - some objects and systems exhibit multiple scaling relationships. It turns out that there exists an entire spectrum of generalized dimensions, D_q where $q \in \mathbb{R}$, of which D_0 corresponds to the box-counting dimension (also known as the capacity dimension) [59]. These generalized dimensions arise from the study of multifractal theory, which involves the study of measures defined over sets. For the purposes of this thesis, it will not be necessary to delve too deep into the specifics of measure theory, so we will just state that a measure μ on a set S is a mapping which assigns real values to any subset $U \subset S$. For example, the measure of $U \subset S$ could be related to the size of U, or it could be related to the probability of visiting U in some stochastic process. Multifractal theory is commonly associated with characterizing complex distributions, such as the probability distribution on a strange attractor, the distribution of growth probabilities on the external surface of a diffusion-limited aggregate, or the spatial distribution of dissipative regions in a turbulent flow [60]. That being said, multifractal behaviour can also arise from a uniform measure defined over a fractal set. In this case we might think of the measure as representing a mass distribution defined over the set, and these sets are sometimes referred to as geometric multifractals, or mass multifractals [61, 62].

The generalized dimensions are closely related to the more well known $f(\alpha)$ singularity spectrum, the hallmark of multifractal theory. Very briefly, the $f(\alpha)$ spectrum provides a description of a multifractal measure both locally and globally by partitioning the support of the measure into subsets S_α which have local dimension (or singularity exponent) α. The value $f(\alpha)$ is the Hausdorff dimension of S_α for each non-empty subset S_α. When the S_α are non-empty and fractal over a range of exponents α, we have a multifractal measure. On the other hand, if we think back to the simpler fractals sets described previously, such as the Cantor set and von Koch curve, we can define a uniform measure over the set (i.e. a uniform mass distribution) and it is clear that the $f(\alpha)$ spectrum will consist of a single point with both α and $f(\alpha)$ equal to the Hausdorff dimension of the set. This type of a fractal set is called a monofractal.

Another way of characterizing the scaling behaviour of a multifractal measure is to compute the generalized fractal dimensions, D_q, which correspond to the scaling exponents of the q^{th} moments of the measure [60]. The generalized dimensions are related to the aforementioned $f(\alpha)$ spectrum by a Legendre transform, and are often easier to evaluate for measures arising from empirical data. If we consider a measure μ defined on $E \subset \mathbb{R}^n$, cover E with $N(\varepsilon)$ boxes of size ε, and let μ_i be the value of μ over the i^{th} box, then we

have the following definition.

Definition 5.2.9. *The spectrum of generalized fractal dimensions, D_q for $q \in (-\infty, \infty)$, is given by*

$$D_q = \frac{1}{q-1} \lim_{\varepsilon \to 0^+} \frac{\log\left(\sum_{i=1}^{N(\varepsilon)} \mu_i^q(\varepsilon)\right)}{\log(\varepsilon)} \tag{5.19}$$

for all $q \in \mathbb{R}$.

In general it can be shown that

$$D_q \geq D_{q'} \text{ for } q' > q \tag{5.20}$$

with equality if and only if the measure is monofractal [59]. Therefore we can consider a set to be multifractal if D_q is a decreasing function of q. Figure 5.8 shows the spectrum of generalized dimensions, D_q, computed for two images; the Sierpinski triangle, a monofractal, in (A), and a diffusion-limited aggregation (DLA) cluster in (B). DLA clusters are generally considered to be multifractal structures [63]. We will discuss DLA clusters further later in this work as they are often compared to vascular networks on the basis of having similar "fractal dimensions". We can see that for the Sierpinski triangle the values of D_q are reasonably constant, whereas for the DLA cluster they decrease with q. The D_q values in Figure 5.8 were computed using an implementation of the generalized sandbox method in Matlab, the details of which can be found in Section 7.3.1.

In this thesis we will not concern ourselves very much with the theory of multifractals, however we will investigate the estimation of the generalized fractal dimension for certain special values of q. In particular, we will see that the dimension D_2, also known as the correlation dimension, is often used interchangeably with the box-counting dimension, D_0, as an alternate means of estimating the "fractal dimension" of a set. Once again, we stress that it is important to distinguish between the different definitions of the various fractal dimensions - they can, and often do, have different values.

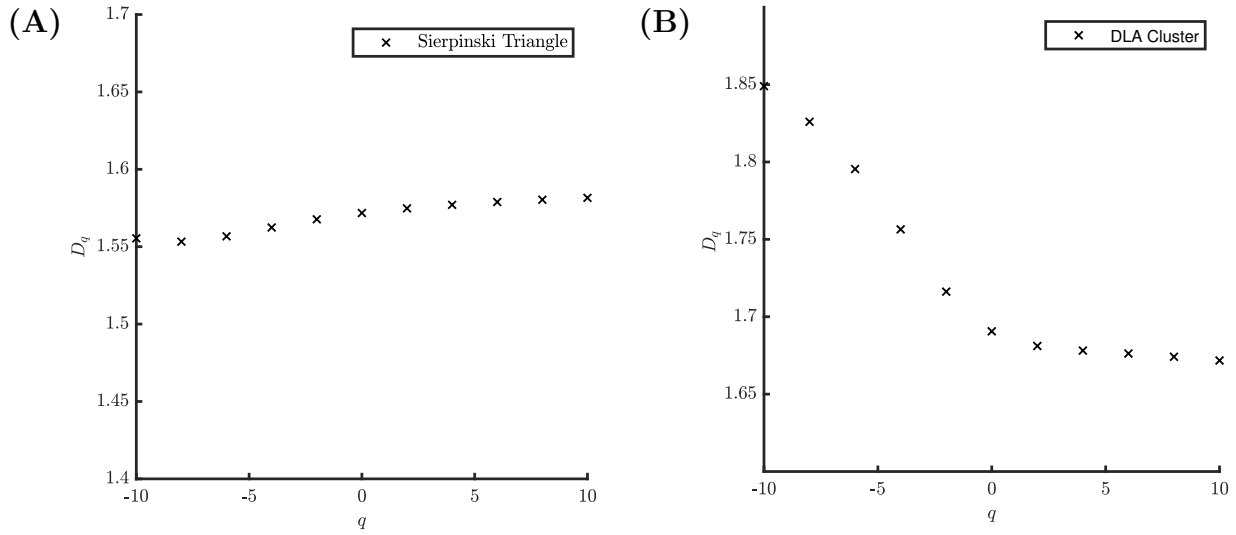

Figure 5.8: The spectrum of generalized dimensions computed for two fractal objects: **(A)** the Sierpinski triangle, a monofractal, and **(B)** a DLA cluster, a multifractal.

Chapter 6

Methods of Estimating Fractal Dimensions

A large variety of natural objects are thought to have fractal properties. For instance, in Mandelbrot's famous book from 1983, *The Fractal Geometry of Nature*, he claims that

> *... [c]louds are not spheres, mountains are not cones, coastlines are not circles, and bark is not smooth, nor does lightning travel in a straight line.* [49] (p. 1).

At the time, Mandelbrot's ideas on fractal geometry and its applications to the natural world were novel, however, now a quick search reveals a plethora of articles on the fractal properties of various natural objects - from clouds and mountains, to stream networks and vascular networks, the latter being the subject of this thesis. Of course, none of these natural objects can possibly be true fractals due to their finite nature, however, they may appear to be fractal over certain ranges of scales. As such, fractal geometry is often used as a tool to characterize and model such objects. One of the most common ways that fractal geometry has been applied to natural objects is in the estimation of their fractal dimensions as a means to characterize their complexity or level of irregularity.

The two primary methods for estimating fractal dimensions from finite images seen in the literature are the box-counting method (by far the most common), and the sandbox (or generalized sandbox) method. These two methods are often used interchangeably to compute the "fractal dimension" of an image, however as previously mentioned, it is important to remember that there are many ways to define a fractal dimension, and they are not always equivalent. The box-counting method and sandbox method compute two

different dimensions; the box-counting dimension (D_0, or capacity dimension) and the correlation dimension (D_2), respectively. In some cases - when the set is a monofractal - these two definitions are equivalent, however when working with naturally occuring objects, whether the object is monofractal, multifractal, or even if it is fractal at all, is not known at the outset.

In order to estimate the dimensions of naturally occurring objects we work with digital images of said objects. In this thesis we will be concerned with objects and structures which can reasonably be represented by two-dimensional binary images, however the methods described in the following sections can be extended to three dimensions by replacing boxes with cubes. Additionally, the generalized sandbox method described in Section 6.2 can be applied to estimate the multifractal properties of greyscale images, by treating the intensity value of each pixel as a measure defined upon the set.

6.1 The Box-Counting Method

The box-counting method is likely the most well known and commonly used method to estimate the box-counting dimension from binary images. We saw in Section 5.2.3 that, when the limit exists, the box-counting dimension of a set $E \subset \mathbb{R}^n$ is given by

$$\dim_B(E) = \lim_{\varepsilon \to 0} \frac{-\log(N(\varepsilon))}{\log(\varepsilon)}. \tag{6.1}$$

In practice the set E is replaced by a finite image, i.e. a discrete collection of pixels, which lies in the plane. Such an image can, of course, only exhibit detail over a finite range of scales and as such, in the finite case, we cannot expect the scaling relationship to hold in the limit as $\varepsilon \to 0$. Instead, in the finite case, we must locate the relevant range of scales over which the scaling relationship holds. Over this range of scales, the set is covered with a grid and the dimension is estimated from the slope of a log-log plot of $N(\varepsilon)$ (the number of boxes which intersect the set) vs. $1/\varepsilon$.

The idea is, that over this range of scales, we can treat the finite set as an approximation for a (statistically) self-similar set, such that the scaling relationship holds at all scales, and can effectively be extrapolated from this finite region down to the limit as $\varepsilon \to 0$. We note that in the finite case we require a stricter version of the scaling relationship from Equation (5.16). Since we cannot take the limit as ε goes to zero in the finite case, we must have $N(\varepsilon) = c\varepsilon^{-s}$, or at the very least $N(\varepsilon) \approx c\varepsilon^{-s}$, for some finite range of ε values in order to measure the scaling exponent, s, from the slope of the log-log plot. The box-counting

method provides an estimate of the box-counting dimension, not of the finite set itself, but of the fractal set which it approximates [64]. In this thesis we will use the notation D^{bc} to refer to estimates of the box-counting dimension computed using the box-counting method.

In Figure 6.1 we show an example of the log-log plot of $N(\varepsilon)$ vs. ε which arises from the application of the box-counting method to a finite approximation of the Sierpinski triangle. The Sierpinski triangle has a box-counting dimension of $^{\log(3)}/_{\log(2)} \approx 1.585$. From the figure

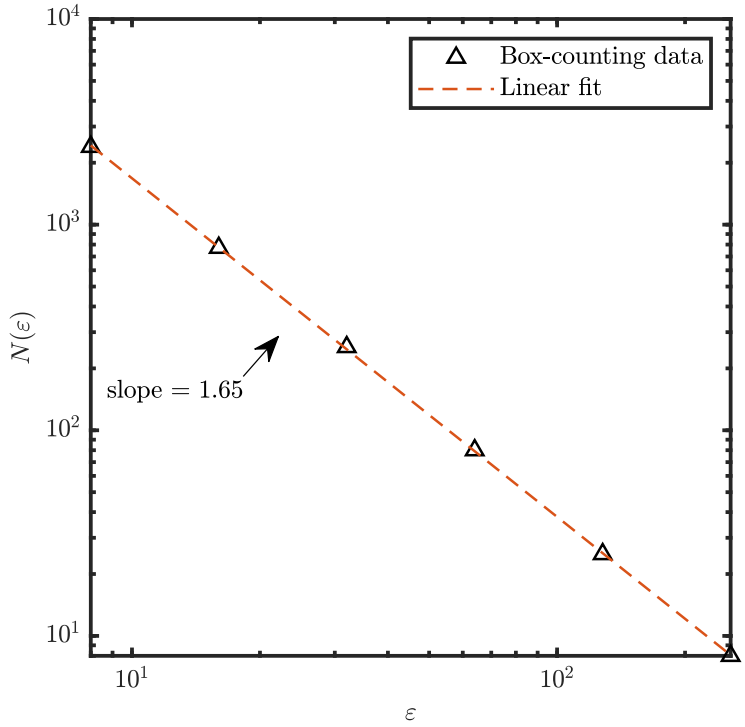

Figure 6.1: A log-log plot of $N(\varepsilon)$ vs. ε computed from a finite approximation of the Sierpinski triangle.

we can see how the slope of a straight line fit through the data points provides a good estimate of the box-counting dimension of the set. This plot was generated using Matlab and a thorough discussion on our implementation of the box-counting method can be found in Section 7.3.1. Despite its popularity and widespread use, the box-counting method is not without its flaws. The algorithm is very sensitive to its parameters, including the box sizes which are used. In Figure 6.1 we knew the dimension of the Sierpinski triangle, so it was easy to find a range of scales over which this behaviour is present in the finite image. In general, however, locating the correct range of scales over which the scaling relationship

holds can be problematic - especially as many of the scaling relationships observed in nature span fewer than two orders of magnitude [65]. This aspect of fractal dimension estimation is mostly glossed over in the literature - it seems that most authors simply select a region of the log-log plot of $N(\varepsilon)$ vs. $1/\varepsilon$ which looks reasonably linear from which to estimate dimension. This method is not particularly accurate for reasons which will be discussed in further detail in Section 6.5 of this thesis.

Another issue with the box-counting method is the quantization error which arises from the use of a grid consisting of finite box sizes. Although we saw in Section 5.2.3 that we get the same result for the box-counting dimension whether we cover the set with the smallest ε-cover or an ε-mesh, the proof of equivalence relies on taking the limit as ε goes to zero. In the finite case, the number of ε-mesh cubes which intersect the set will depend on the location and orientation of the image relative to the grid [64]. In practice the quantization error is often reduced by translating the grid randomly or systematically and taking the minimum value of $N(\varepsilon)$ over multiple grid orientations. It can also be reduced by using the "enlarged box technique", originally proposed in [66] and improved upon in [67]. There will, however, always be some amount of quantization error due to the fact that we are using a finite grid where the boxes are not allowed to shift relative to one another.

Finally, box-counting suffers from what are commonly referred to as "edge effects", referring to the boxes near the border of the set, which may intersect the set in only a minor way. At large scales, these edge effects become increasingly prominent and may lead to deflated estimates of dimension if they are not accounted for properly. Edge effects can be mitigated somewhat by enforcing a minimum density before a box gets counted, or by restricting boxes to the interior of the set, as in [68], however this approach requires quite large sets in order to be practical.

6.2 The Sandbox Method

The sandbox method, as described in [69], is often presented as an alternative method of estimating the fractal dimension from an image. The sandbox method is a means of estimating the correlation dimension, which is often the same, or similar to the box-counting dimension, from finite images. In the sandbox method, occupied pixels are surrounded by a box of size ε and the average mass (i.e. number of pixels) inside a box of size ε is computed by averaging over the boxes surrounding each pixel in the set. Depending on the implementation, the average mass may be computed over boxes surrounding all occupied pixels or a selection of boxes with centres selected randomly according to a uniform

distribution. Once again, we assume a (statistically) self-similar set, and so the average mass, $\overline{M}(\varepsilon)$, is assumed to scale (at least approximately) according to a power law,

$$\overline{M}(\varepsilon) \approx c\varepsilon^s, \tag{6.2}$$

where c is a positive constant. Once again, since we are working with a finite image, this relationship is assumed to hold over some finite range of scales. The scaling exponent, s, is estimated as the slope of the straight line fit to a log-log plot of $\overline{M}(\varepsilon)$ vs. ε.

The sandbox method suffers from the same issue as the box-counting method with regard to the difficulty in locating the linear region of said plot. However, since the boxes in the sandbox method are centred on individual pixels, issues encountered in the box-counting method, e.g., quantization errors caused by the use of a fixed grid, are not encountered. Typically, the sandbox method is more computationally expensive than the box-counting method, however we note that the run time of the sandbox method depends on the density of the image. Thus for sparse images, the sandbox method can actually be quite efficient.

6.3 The Generalized Sandbox Method

The sandbox method, as described above, is actually a special case of the generalized sandbox method described in [61]. The generalized sandbox method is a method for computing the spectrum of generalized fractal dimensions, D_q. We recall from Section 5.2.4 that the definition of the generalized fractal dimensions comes from defining a measure over a set. In Section 5.2.4 and in that which follows, we have assumed a uniform measure defined over the set, such that the value of the measure over each box is analogous to the mass (or number of pixels) contained within the box. According to the generalized sandbox method, the dimension D_q can be estimated as $1/(q-1)$ times the slope of a log-log plot of $\overline{M}^{q-1}(\varepsilon)$ vs. ε. By setting $q = 2$ in the generalized sandbox method we recover what is typically referred to as, simply, the sandbox method, an estimate of the correlation dimension, D_2. If we set $q = 0$ in the generalized sandbox method, the result is another approximation of the box-counting dimension, D_0. In this thesis we will use the notation D_0^{sb} and D_2^{sb} to refer to estimates made using the generalized sandbox method of D_0 and D_2, respectively.

6.4 Other Methods

In addition to the generalized sandbox method for computing the spectrum of generalized dimensions, a number of methods for applying multifractal theory to discrete sets have been developed in the literature throughout the years [70]. The most natural way to estimate the $f(\alpha)$ spectrum, as used in [71], is to directly estimate the singularity exponent at each point x by estimating the slope of $\log(\mu_i(\varepsilon))$ as a function of $\log(\varepsilon)$ where $\mu_i(\varepsilon)$ is the amount of measure (or mass) in a box of size ε surrounding the i^{th} pixel. The dimension of the set of points with singularity exponent α, i.e. $f(\alpha)$, is then estimated using the box-counting method. A similar method of directly estimating the $f(\alpha)$ spectrum is called the histogram method [72]. The set is covered with boxes of size ε and for each box we compute the exponent

$$\alpha_i = \frac{\log(\mu_i(\varepsilon))}{\log(\varepsilon)} \tag{6.3}$$

where, again, $\mu_i(\varepsilon)$ is the amount of measure (or mass) in the i^{th} box. If $N_\alpha(\varepsilon)$ is the histogram of the values $\{\alpha_i(\varepsilon)\}_i$, $f(\alpha)$ can be estimated from estimates of the power law scaling behaviour of $N_\alpha(\varepsilon)$. In other words, the slope of $\log(N_\alpha(\varepsilon))$ as a function of $\log(\varepsilon)$ is used to estimate $f(\alpha)$.

In [60] the authors propose a different method of computing $f(\alpha)$ for a multifractal measure $P(x)$ by constructing a one-parameter family of normalized measures $\mu(q)$ over the set such that the value of $\mu(q)$ in the i^{th} box of size ε is given by

$$\mu_i(q,\varepsilon) = \frac{(P_i(\varepsilon))^q}{\sum_j (P_j(\varepsilon))^q}. \tag{6.4}$$

The Hausdorff dimension of the support of $\mu(q)$ is related to the entropy and we can compute

$$f(q) = \lim_{\varepsilon \to 0} \frac{\sum_i \mu_i(q,\varepsilon) \log(\mu_i(q,\varepsilon))}{\log(\varepsilon)} \tag{6.5}$$

where $\mu_i(q,\varepsilon)$ is the value of $\mu(q)$ in the i^{th} box of size ε. Similarly, the average value of the singularity strength with respect to the measure $\mu(q)$ can be computed as

$$\alpha(q) = \lim_{\varepsilon \to 0} \frac{\sum_i \mu_i(q,\varepsilon) \log(P_i(\varepsilon))}{\log(\varepsilon)}. \tag{6.6}$$

Equations (6.5) and(6.6) provide an alternative definition of the singularity spectrum. Once again, the limiting values as $\varepsilon \to 0$ are estimated based on the slopes of the relevant log-log plots over some finite range of scales.

Finally, in [72] the wavelet transform modulus maxima (WTMM) method for estimating the $f(\alpha)$ spectrum of multifractal functions was introduced. In this method the singularity spectrum is determined from estimates of the scaling behaviour of partition functions defined based on the modulus maxima of the wavelet transform. This method can also be extended to two-dimensional multifractal measures using the two-dimensional wavelet transform, as was demonstrated in [73].

We shall not discuss these other methods of estimating fractal dimensions in further detail in this thesis. Instead, our discussion will focus on the box-counting and sandbox methods which are used with more regularity in the literature pertaining to fractal dimension estimation of naturally occurring branching structures. We will, however, make an important comment here, which is to draw attention to what all of the methods discussed in this section have in common. Each method described above is based on the estimation of some limiting value (the exponent of a power law scaling relationship specifically) by using linear regression to estimate the slope from a logarithmic plot over a finite range of scales. This means that, just as with the box-counting and generalized sandbox methods, the scaling behaviour we are attempting to measure must be present and must hold reasonably constant at finite scales. Consequently, the discussion which follows in relation to the box-counting and generalized sandbox method applies to the methods described in this section as well.

6.5 Local Slopes

One of the main problems with the usual methods of estimating fractal dimensions from finite images, is the difficulty in determining the optimal range of scales over which to measure their scaling behaviour. When dealing with finite objects, they can, and often do, display different scaling behaviour over different ranges of scales. An excellent example of this phenomenon is illustrated in the video *Fractals are typically not self-similar* at the 15:40 mark [74]. A thin tube-like object is shown which appear one-dimensional when viewed at large scales (low resolution), i.e. its mass scales according to length. However as we zoom in, further detail is revealed and we can see that the surface of the tube is actually two-dimensional. Finally, if we zoom in even further to a very fine resolution, what looked like a surface is revealed to be a tightly wound helical coil, which is actually one-dimensional. So what dimension is this object? If we attempt to estimate its dimension using either the box-counting or sandbox methods, we will observe each of these scaling

behaviours over different ranges of box sizes. We might think that we should look at the smallest box sizes in accordance with the theoretical definition of dimension, however this may not always capture the scaling behaviour we are interested in measuring.

As we have already discussed, naturally occurring objects may have fractal scaling properties over some finite range of scales, and we can assign them a fractal dimension by extrapolating this behaviour down to infinitesimally small scales, i.e. treating them as approximations of strictly self-similar (or statistically self-similar) objects. Since the behaviour we are interested in occurs over a finite range of scales, there will always be some lower cut-off size at which point the scaling behaviour ceases to be present. At such small scales, the object will appear to scale according to the dimension of its smallest constituent part (whether that be individual pixels, line segments, or small two-dimensional regions). If we were to use this smallest range of box sizes to estimate the dimension of our object, we would of course measure the dimension of this smallest constituent part (which will always be an integer). This is not incorrect, this is the true dimension of the finite object, however it does not tell us anything about the scaling behaviour we were interested in measuring. We note here that there will also be some upper cut-off size at which the scaling behaviour will change. At large enough scales, any finite object will appear to be zero-dimensional as the entire mass of the object fits into one single box [65].

As an example, in Figure 6.2 we present two images which approximate the Sierpinski triangle, one of the fractals discussed in Section 5.1. Both images have been generated by the repeated application of a set of contraction mappings - i.e. an iterated function system (IFS). In (A) the procedure is iterated four times, leading to a very rough approximation of the Sierpinski triangle, while in (B) the IFS is applied twenty times, yielding a very good approximation. Clearly if we analyze the image in (A) using very small box sizes, the image will appear one-dimensional, since we will resolve the individual line segments of which it is made up of. On the other hand, the image in (B) displays the scaling behaviour of the Sierpinski triangle down to a resolution of just a few pixels.

What this means is that in order to estimate the scaling behaviour we are interested in, we need to first determine the range of scales over which it is present. It turns out that this is actually quite difficult. In much of the literature, this region is determined by looking directly at the log-log plot of either $N(\varepsilon)$ v.s. ε (box-counting method) or $\overline{M}(\varepsilon)$ v.s. ε (sandbox method) and choosing a region which looks reasonably linear. The r^2 value (coefficient of determination) is often used to confirm the goodness of the linear fit. Unfortunately, as has been pointed out in [75, 65, 64], the independence assumption of linear regression does not hold for the box-counting or sandbox methods. The quantity $\overline{M}(\varepsilon)$ or $N(\varepsilon)$ at a given box size depends on the value at previous sizes and consequently, when the slope is estimated using regression, the r^2 values are inflated and may give a false

(A) **(B)**

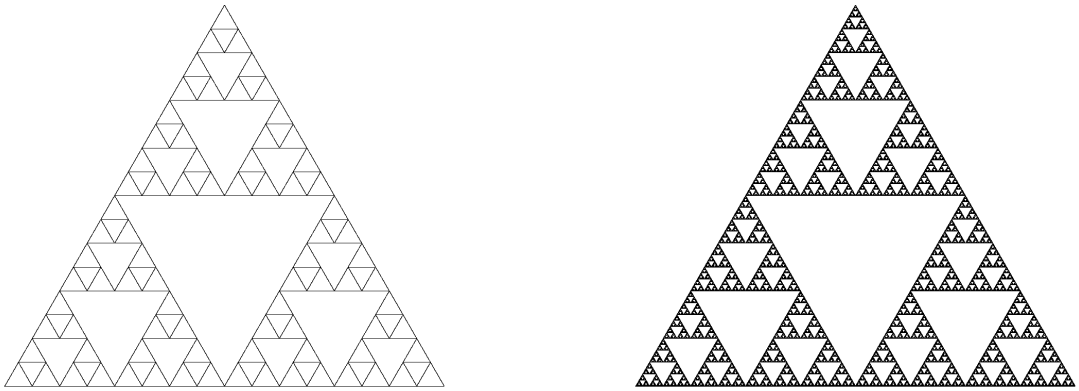

Figure 6.2: **(A)** A very rough approximation of the Sierpinski triangle, starting with the outline of an equilateral triangle the recursive process is repeated just four times. **(B)** A better approximation with twenty iterations.

sense that the underlying data is linear even when it is not.

Something which has been proposed in the past as a better method for finding the linear regions of the log-log plots used in fractal dimension estimation [68, 76], but has not generally been adopted in the recent literature, is to compute the local slopes of these log-log plots over small ranges of ε values. In other words, to apply linear regression and estimate the slope over a small discrete set of ε values which satisfy $\varepsilon_1 \leq \varepsilon \leq \varepsilon_2$, as shown in Figure 6.3. In this example we have illustrated the local slopes over two distinct regions consisting of three ε values each. We see that the local slopes vary with ε, indicating that the scaling behaviour is not constant over the range of scales pictured.

In Section 7.3.1 we will discuss further the choice of these ε values and how that may impact the results. Analysing the local slopes may provide us with a much clearer picture of how the scaling behaviour of an object changes with the scale (or resolution) we use to measure it. We can even plot the local slopes as a function of ε and use these plots as a tool to identify regions of constant scaling behaviour. Figure 6.4 shows one such plot.

In Figure 6.4 we have plotted the local slopes resulting from the box-counting method applied to each of the approximations of the Sierpinski triangle shown in Figure 6.2. The local slopes have been plotted as a function of the upper box size, ε_2. We can see that in both cases the local slopes settle around the theoretical dimension of the set, however

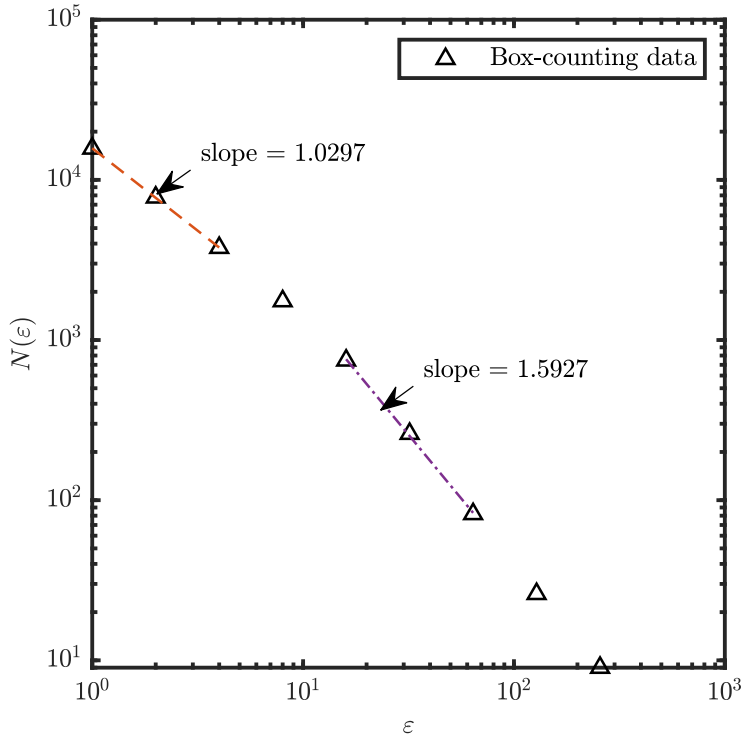

Figure 6.3: Computing the local slopes from a plot of $N(\varepsilon)$ vs. ε. Here, the local slopes are computed for the image in in Figure 6.2 (A).

the range of scales over which we observe this behaviour changes. This plot clearly shows us that there is lower cut-off scale below which Figure 6.2 (A) appears one-dimensional, whereas in Figure 6.2 (B) the fractal scaling relationship is present down to very small scales. More details on how these plots were generated can be found in Section 7.3.1.

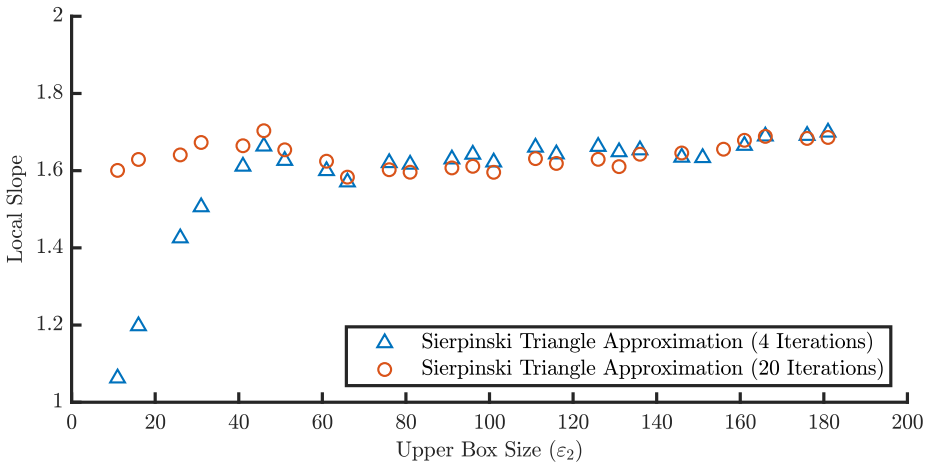

Figure 6.4: A plot showing the local slopes resulting from the box-counting method for the two finite approximations of the Sierpinski triangle shown in Figure 6.2.

Chapter 7

Fractal Trees (and Other Branching Structures) are Not Self-Similar

As previously mentioned, estimates of fractal dimension have been used as a tool to characterize all sorts of natural images over the years. In this thesis we are concerned specifically with the estimation of the fractal dimensions of a particular class of objects, namely branching structures, which are of particular interest for two reasons:

1. Branching patterns are ubiquitous in nature. For example we observe branching in trees, rivers/streams, the vascular system, branching airways in the lungs, etc...

2. Due to the process by which they are generated, branching structures are not strictly self-similar, yet they can still have fractal properties. Fractal trees, for instance, have been shown to have a Hausdorff dimension which is greater than their topological dimension in certain cases [49].

By means of a careful analysis of both the box-counting and sandbox methods, we will show that the lack of strict self-similarity (or non-homogeneity) of fractal trees and other branching structures causes the usual methods for estimating fractal dimensions to fail. Despite this, one can find many studies which use these methods to estimate the fractal dimensions of various branching structures, such as stream networks [77, 78], root systems [64], and vascular networks [79]. In these studies, fractal dimension is used to characterize the complexity of branching patterns. Specifically, in the case of vascular networks, as we discuss in detail in Section 7.1 below, the results have been found to distinguish between healthy and diseased morphology, and thus used to draw conclusions about the fractal

processes generating such networks. We propose that results such as these should be treated with some skepticism as the methods used to estimate dimension rely on an assumption of self-similarity which branching structures do not satisfy.

We also point out here that we are not alone in our questioning of the indiscriminate and widespread application of fractal dimension estimation methods to images of natural objects. In 1995, as the application of fractal theory to all sorts of naturally occurring structures was becoming prevalent, the authors of [68] suggested a simpler explanation for the branching patterns observed in retinal neurons which had previously been hypothesized to be fractal. They showed, using a systematic approach, that such structures are space-filling, not fractal. Later that same year, J.D. Murray cautioned against the use of fractal measurements to draw conclusions about biological processes, noting that

> *The problem with a good name for a new field is that [...] inappropriate use can raise unrealistic expectations as to its relevance and applicability. This is particularly true for fractal theory which can be visually dramatic and can be practised without much background or sophistication.* [80] (p. 369).

Adding further support to the criticisms of fractal theory, in 1996, the authors of [81] demonstrated that random distributions exhibit apparent fractal behaviour over a range of scales consistent with the typical range observed in experimental measurements of fractal objects. These early words of warning were not generally heeded, and studies on the estimation of the fractal dimensions of natural objects have continued to appear. Amidst these studies, some more recent criticisms have emerged including [65], in which the authors discuss some of the difficulties in applying fractal methods to ecological data, concluding that evidence of a scaling relationship which spans only a few orders of magnitude is not sufficient evidence for true fractality, and [64], which cautions against computing fractal dimensions of root systems without first rigorously testing for self-similarity, or statistical self-similarity.

Before we continue, it should be noted here that the comments in the following sections extend to multifractal methods as well since the determination of both the generalized fractal dimensions and the $f(\alpha)$ curve depends on least-squares fitting of box-counting data and thereby suffers from the same problems we will discuss.

7.1 A Brief Review of the Literature on the Fractal Dimensions of Vascular Networks

The primary motivation behind this work is the large volume of literature which exists pertaining to the fractal dimensions and fractal properties of vascular networks. The blood vascular system in humans and other animals is comprised of arterial and venous trees, hierarchical branching structures which span a large range of scales and which are connected at their extremities by the capillary network, the smallest vessels. Given their complex and hierarchical structure, it is not surprising that many researchers in the medical sciences have attempted to characterize vascular networks using fractal dimension and other fractal measures. Fractal tree models have even been used to generate realistic simulations of vascular networks [82]. Furthermore, a number of pathological conditions, including cancers and degenerative diseases, lead to changes in vessel morphology and branching structure. Tumour vasculature, for example, is known to be more chaotic in appearance than healthy vasculature. The identification of tumour angiogenesis as a potential target for the treatment of certain cancers has motivated many researchers to use fractal approaches to try to understand the mechanisms by which tumour vasculature forms [79]. Fractal properties, including fractal dimensions, have also been proposed as a tool to quantify both the growth [83] and normalization [69, 84] of tumour vasculature.

In [85] and [69] the authors use both the box-counting and sandbox methods to measure the fractal dimension of healthy and tumour vascular networks grown in mice bearing dorsal skinfolds. A selection of sample images from the study in [85] are shown in Figure 7.1 where we observe that healthy arteriovenous networks have a clear tree-like structure, healthy capillary networks have a more uniform, grid-like appearance, and tumour networks, in which arteries, veins and capillaries cannot be distinguished, appear more disordered. The authors find good agreement between the box-counting and sandbox methods. Their measurements show that the dimension of healthy vasculature is in agreement with that of diffusion-limited aggregation (approximately 1.71), while the dimension of tumour vasculature agrees with that of critical percolation clusters (approximately 1.896), and capillary networks are found to have a dimension in the range $1.96 - 2$, which is consistent with that of a 2-dimensional object or a space-filling curve in two dimensions. They claim that the result for healthy vasculature is in agreement with the accepted view of the angiogenic process, whereas the results for tumour vasculature are used to form the basis of a novel hypothesis that tumour angiogenesis is a local growth process. Additional results in [69] corroborate these results, additionally showing that anti-angiogenic treatments applied to tumour vasculature lead to decreased estimates of fractal dimension.

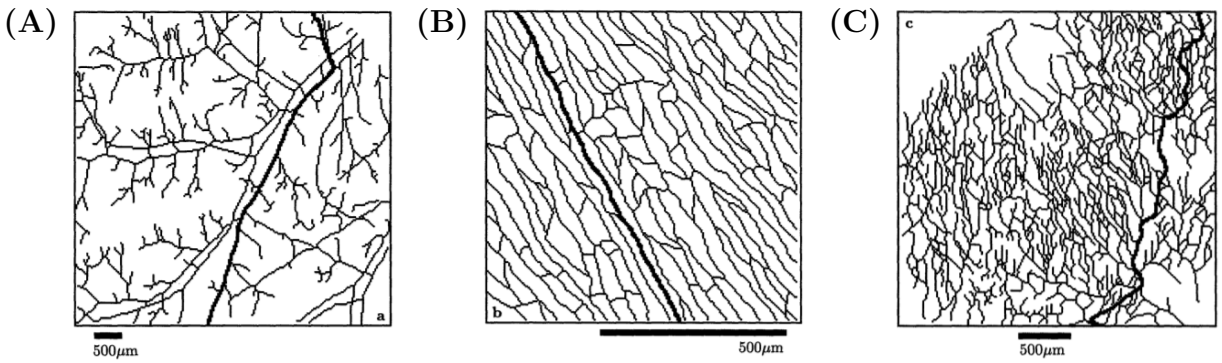

Figure 7.1: Representative skeletonized images of three types of vascular networks. Bolded lines represent the minimum path for each network. **(A)** Healthy subcutaneous arteriovenous network. **(B)** Healthy subcutaneous capillary network. **(C)** Tumour network. **Note:** Figure reproduced from [86] with permission.

A number of further studies have been performed on both experimental images of healthy and tumour vasculature [87, 84, 88] as well as simulations of tumour vasculature [89, 83]. These studies, performed on different data sets, all tend to agree on the fractality of both healthy and tumour vasculature, and when compared, tumour vasculature is consistently found to have a greater dimension than healthy arteriovenous networks. That being said, the specific numerical estimates of dimension across studies are somewhat less consistent. Given the inconsistencies in imaging methods, image processing methods and in the implementations of fractal dimension estimation methods, this is not surprising. One simply has to look at Table 1 in [76] for an example of the variation in estimates of the fractal dimension of healthy vasculature over the years.

Fractal dimension estimation methods have also been widely applied to imaging of retinal vasculature, perhaps even more so than tumour vasculature, due to the ease of high quality image acquisition. A number of studies have found the dimension of healthy retinal vasculature to be consistent with that of diffusion limited aggregates (DLA clusters) [90, 91, 92, 93], in line with what was found in [85] for healthy arteriovenous networks in mice. Despite this, a fairly recent work [94] has reported much lower dimensions on a fairly large database of retinal images. Some studies have also compared the retinal vasculature of cognitively healthy patients to that of patients with degenerative diseases, finding that estimates of dimension are lower for cognitively impaired patients [94, 95].

We make note here of a general observation that the results in the literature tend to find that estimates of fractal dimension correlate positively with vessel density. For example, as we saw in Figure 7.1, tumour networks appear much more dense than healthy vasculature,

and capillaries, occurring at the smallest scales, have the highest vessel density. In [96] and [97] a positive relationship between vessel density and estimated fractal dimension is shown directly. A correlation between density and dimension is perhaps expected, since fractal dimension is often considered a measure of how space-filling an object is. However, it turns out that in the case of branching structures, estimates of fractal dimension are directly influenced by the spacing between nearby branches, and consequently, the vessel density. We will see why this is the case in the following sections.

7.2 On the Self-Similarity (or Lack Thereof) of Fractal Trees and Other Branching Structures

Previously, we discussed fractal trees as an example of a class of mathematical objects which are generally considered to be fractals despite not being strictly self-similar. More generally, let's consider the class of all branching structures, which includes but is not limited to fractal trees. For the purposes of this work we will consider branching structures to be objects which are generated by repeated branching (i.e. application of a branching generator as described in Section 5.1.2) of a trunk (an infinitely thin line segment) into scaled and rotated copies of itself. We will not require the same branching generator to be applied on every iteration of branching, however we will require that $N \geq 2$ on at least some iterations of the process to avoid the trivial case of a straight (or bent) line being considered as a branching structure. We will also only consider trees which do not have branches with significant overlap, as such trees may be completely space-filling, i.e. two-dimensional. We make note here of the fact that grids and grid-like structures fall into our definition of branching structures since we have not placed any restrictions on self-contacting branches.

Although some branching structures may have fractal properties, the very nature by which they are generated precludes them from being self-similar. According to our definition of strict self-similarity from Section 5.1.1, in order for a set to be strictly self-similar any arbitrary part of the set must contain a copy of the whole set. If we consider an arbitrary branching structure and select a portion of the set which lies on the trunk, then this subset will be a straight line. Since we have specified $N \geq 2$ on at least some iterations of the branching procedure, it is not possible that this portion of the set contains a copy of the whole set, as it contains no branches. As we saw in our previous discussion on fractal trees, some parts of the set may contain scaled copies of the whole, however it is not possible for a branching structure to be truly self-similar. Moreover, the non-homogeneity of fractal trees and general branching structures violates the self-similarity assumptions of the box-counting method and multifractal approaches such as the generalized sandbox method and the other

methods discussed in Section 6.4. As we will see, this directly impacts our ability to use these methods to estimate their dimensions from finite approximations.

Realistic branching structures are, of course, not infinitely thin. In order to focus on the structural information, images of branching structures are typically skeletonized prior to the estimation of their fractal dimensions or other properties. In very simple terms, the skeletonization process reduces a binary image to what is called its skeleton - a minimal set of pixels which preserves the shape and connectivity of the original image. In the case of branching structures, the skeleton is typically a connected set comprised of line segments, like those shown in Figure 7.1 in the previous section. Some works do consider the diameters of the branches in their analysis, however a similar argument shows that such "thick" branching structures are also not self-similar. As a result, similar problems in estimating the fractal dimension are found to persist regardless of whether images are skeletonized or left in their original form. For simplicity, we will deal solely with skeletonized branching structures in this work, noting that the discussions which follow apply more generally to "thick" branching structures as well.

Before moving on, we reiterate that despite their lack of self-similarity, some trees are in fact fractals and have a non-integer Hausdorff dimension. One way we can make sense of this is to consider a fractal tree as the union of two parts. One part consists of the trunk and the branches, and has dimension one, which is equal to its topological dimension. The second part consists of just the tips of the branches - the so-called tree "canopy" - which is strictly self-similar with dimension D. When $D > 1$, the scaling behaviour of the canopy is dominant in the limit and the dimension of the whole tree is equal to D [49]. A derivation of the box-counting dimension of a simple fractal tree which clearly illustrates this phenomenon is presented in Appendix A. From the discussion above, one might assume that measurements of the dimension of fractal trees would be underestimates of the true dimension (as a result of the tree canopy not necessarily dominating in the finite case). However, it turns out that the opposite is true due to transitions in the scaling relationship which are observed at characteristic box sizes related to the density of branches in the image.

7.2.1 Theoretical Analysis Using Local Slopes

In this section we will investigate how the lack of self-similarity of fractal trees and other branching structures impacts the estimates of their dimensions resulting from traditional methods like the box-counting and generalized sandbox methods. Let's consider a binary image which depicts a branching structure. Since we are considering a finite image this is,

of course, at best a finite approximation of some infinite branching process, and as such is made up of a finite number of finite line segments which form a connected set. Away from the edges of the set (i.e. ignoring any edge effects), the mass in a box of size ε can always be written as $A\varepsilon$, where A is some positive constant related to the number of line segments which pass through the box and the angle at which they pass through. If A is constant over some range of scales, then the object scales as a one-dimensional object, and this will be reflected in the results of either the box-counting or sandbox methods.

Now, consider the image of a tree depicted earlier in Figure 5.6. If we cover the entire image with very small boxes, we expect that most boxes will contain a single straight line. Now imagine covering the set with slightly larger boxes - a good number of boxes will contain two (or more) line segments due to branches which are sufficiently close to each other so as to fit into a single box. This results in a change in the mass-scaling relationship which violates the self-similarity assumption of both the box-counting and generalized sandbox methods. We can use the framework of the generalized sandbox method in order to gain some insight into how these transitions in the mass-scaling relationship directly affect the results of estimating both the box-counting and correlation dimensions. We recall here that the generalized sandbox method can be used to estimate all of the generalized dimensions, including both D_0 (the box-counting dimension) and D_2 (the correlation dimension). Specifically, we will calculate the local slopes of the log-log plot of $\overline{M}^q(\varepsilon)$ vs. ε between pairs of consecutive ε values and investigate how these local slopes change near a point of transition.

Consider an image, $\mathbf{I}$, comprised of P pixels in total and denote the mass in a box of size ε centred on the i^{th} pixel by $M_i(\varepsilon)$. For simplicity, we assume that ε is small enough relative to the size of the image that we can ignore the edge effects which result from the usual computational methods. We assume that $\mathbf{I}$ has some characteristic size ε^* such that when $\varepsilon < \varepsilon^*$ the mass-scaling relationship is constant everywhere, say $M_i(\varepsilon) = A\varepsilon$, and when $\varepsilon > \varepsilon^*$ the scaling relationship in some fraction of the boxes transitions to a new scaling relationship, $M_i(\varepsilon) = B\varepsilon$. Note that $A, B \geq 0$. Now, consider two box sizes ε_1 and ε_2 which satisfy $\varepsilon_1 < \varepsilon^* < \varepsilon_2$. We can compute the average mass in a box of size ε_1 as

$$\begin{aligned} \overline{M}(\varepsilon_1) &= \frac{1}{P}\sum_{i=1}^{P} M_i(\varepsilon_1) \\ &= \frac{1}{P} \cdot P \cdot A\varepsilon_1 \\ &= A\varepsilon_1. \end{aligned} \tag{7.1}$$

Letting T denote the fraction of boxes in which the scaling relationship transitions, we

compute the average mass in a box of size ε_2 to be as follows,

$$\begin{aligned}\overline{M}(\varepsilon_2) &= \frac{1}{P}\sum_{i=1}^{P} M_i(\varepsilon_2) \\ &= \frac{1}{P}\left(T \cdot P \cdot B\varepsilon_2 + (1-T) \cdot P \cdot A\varepsilon_2\right) \\ &= T \cdot B\varepsilon_2 + (1-T) \cdot A\varepsilon_2.\end{aligned} \tag{7.2}$$

The dimension is estimated by computing the slope of the log-log plot of $\overline{M}^{(q-1)}$ v.s. box size (ε). We set $q = 2$ (for the correlation dimension) and compute the local slope between the two points ε_1 and ε_2, given by

$$\begin{aligned}\Delta_2 &= \frac{\log(\overline{M}(\varepsilon_2)) - \log(\overline{M}(\varepsilon_1))}{\log(\varepsilon_2) - \log(\varepsilon_1)} \\ &= \frac{\log\left(\frac{T \cdot B\varepsilon_2 + (1-T)\cdot A\varepsilon_2}{A\varepsilon_1}\right)}{\log\left(\frac{\varepsilon_2}{\varepsilon_1}\right)}\end{aligned} \tag{7.3}$$

For simplicity, we let $m = \frac{\varepsilon_2}{\varepsilon_1}$ so that

$$\begin{aligned}\Delta_2 &= \frac{\log m}{\log m} + \frac{\log(T \cdot B/A + (1-T) \cdot A/A)}{\log m} \\ &= 1 + \frac{\log(1 + T \cdot (B-A)/A)}{\log m}.\end{aligned} \tag{7.4}$$

A similar argument can be made, setting $q = 0$, to compute the local box-counting dimension between box sizes ε_1 and ε_2 as

$$\Delta_0 = 1 - \frac{\log(1 + T \cdot (A-B)/B)}{\log m}. \tag{7.5}$$

From Equations (7.4) and (7.5), we can see that the local slope will be greater than 1 if B is greater than A, and less than 1 if A is greater than B. The impact of the transition on the local slope, unsurprisingly, is dependent on T, the fraction of pixels which transition and, somewhat surprisingly, also depends on m, the ratio of the box sizes used in the algorithm. If we increase m, i.e. use box sizes which are spaced further apart, the impact of the transition on the local slope will be diminished. The results presented in this section are specific to the generalized sandbox method, however we expect similar effects to be

observed in the local slopes when applying the box-counting algorithm as well. It is not possible to write down the exact local slopes for the box-counting algorithm since the minimum number of boxes needed to cover an image depends not only on the average mass distribution, but also the spatial locations of individual pixels. That being said, the total mass of the image divided by the average mass in a box of size ε is a good approximation of the number of boxes required to cover the image in most cases, and so the results for Δ_0 should roughly apply to the box-counting algorithm as well.

7.3 Computational Results

In this section we will present some computational results which illustrate the observations of the previous section. In order to do so we have implemented both the box-counting and generalized sandbox methods and directly computed the local slopes over small ranges of box sizes for each of the estimated dimensions D^{bc}, D_0^{sb} and D_2^{sb}. First, we will directly validate the results from Section 7.2.1 using some very simple images, then we will see how the same phenomenon extends to more complex branching structures by analysing images of both retinal vasculature and simulated fractal trees.

7.3.1 Implementation Details

For the purposes of analysing the resulting local slopes, both the box-counting and generalized sandbox methods have been implemented in Matlab according to the descriptions in Sections 6.1, 6.2 and 6.3. Specifically, in our implementation of the generalized sandbox method we used boxes centred on all occupied pixels in order to compute the average mass. The publicly available FracLac software [98], a popular tool for the estimation of fractal dimensions, was used as a baseline for developing our implementations of both methods. Both methods were validated on a selection of well-known fractal images with good results.

Accuracy and Computational Cost

As might be expected, when both methods are implemented on images of standard fractals we observe that the generalized sandbox method is generally more robust than the box-counting method. Unfortunately, this increase in accuracy often comes at the expense of higher computational cost. The run time of the box-counting method depends on the image size, whereas the run time of the sandbox method depends on the number of foreground

pixels in the image, i.e. both image size and image density. Typically this means that the generalized sandbox method is quite a bit slower slower, however we note that when working with very sparse images the generalized sandbox method may actually run faster than the box-counting method. We note that the run time of the sandbox method can also be improved by computing the average mass over a smaller selection of random centres, as has been done with good results, for example, in [62].

Selection of Box Sizes

Both the box-counting and generalized sandbox methods rely on selecting a set of box sizes over which to tabulate and fit the data. As we have already seen in our discussion regarding local slopes, the selection of these box sizes can have a significant impact on the results. In our implementation we allow for the selection of both a minimum and maximum box size. The minimum box size is selected in pixels, and should be larger than the smallest component of the image, whereas the maximum box size is selected as a percentage of the total image size. For this purpose, the image size is defined as the minimum of the width and height of the smallest rectangle which contains the foreground pixels of the image. It is generally accepted that box sizes beyond about 50% of the image size should not be used as they do not capture the scaling behaviour of the image accurately [98]. In addition to setting the minimum and maximum box sizes, we also need to select which box sizes to use within that range. The box sizes may be selected either according to a power law or incremented linearly. Figure 7.2 shows the difference between these two options. When we use power law scaling the box sizes are uniformly spaced on a log-log plot, meaning that all scales are evenly weighted in the linear fit, however the gaps between the larger box sizes can be quite large. This means that if the scaling behaviour we are looking for only occurs over a small range of scales it might be completely missed. For that reason, it is more common to use linearly incremented box sizes when applying the box-counting or generalized sandbox method to real data where the scaling relationship is unknown at the start. One drawback is that when we are dealing with larger box sizes, a small linear change in the box size may not always cause a change in the value of $N(\varepsilon)$ (or $\overline{M}(\varepsilon)$), which can lead to flat regions in the plot. To compensate for this, we apply a smoothing filter the resulting data. The FracLac software [98] provides two different options for smoothing the data, in our implementation we use what they call the "small" smoothing filter. Essentially, for a given count (either $N(\varepsilon)$ or $\overline{M}(\varepsilon)$) we keep the smallest box size which has that count and remove all other data points.

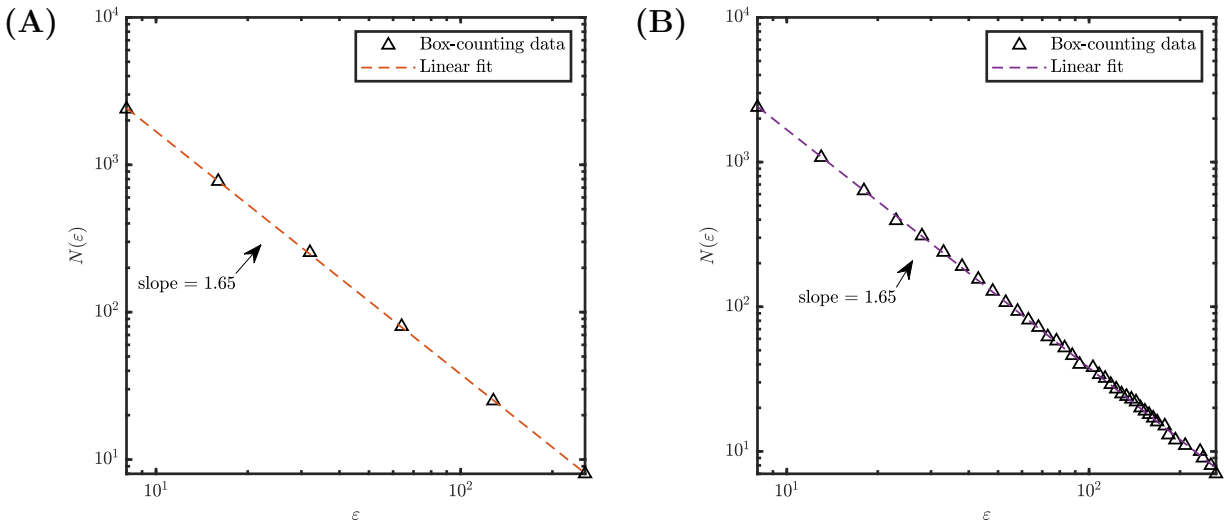

Figure 7.2: Two log-log plots of $N(\varepsilon)$ vs. ε resulting from the box-counting method applied to a finite approximation of the Sierpinski triangle. In **(A)** the box sizes are chosen according to a power law with base two, whereas in **(B)** the boxes are incremented linearly with a spacing of five pixels.

Reducing Quantization Error and Edge Effects

To reduce the quantization error which results from the use of a finite fixed grid in the box-counting method, we employ 100 random translations of the grid. The smallest value of $N(\varepsilon)$ is taken over all of these grid orientations. In the generalized sandbox method, we take the average mass over boxes centred on the foreground pixels of the image, so no such grid translation is needed.

Both methods are susceptible to edge effects, especially at larger box sizes where boxes near the edge of the image may contain only a few pixels. One way of dealing with this is to use reasonably small box sizes (less than 50% of the image size as previously mentioned), however we can also set a minimum density and require that each box contains at least that density of pixels in order to be included in the count. We have set this parameter to 10% of the overall image density in order to somewhat mitigate edge effects without discounting too much of the image.

Computing the Local Slopes

Although our implementations of the box-counting and generalized sandbox methods could be used to compute a single number for a "fractal dimension", that is not their main purpose. Instead, we use the results of these programs to estimate the local slopes over small subsets of the data, as described in Section 6.5. The method used to compute the local slopes varies slightly depending on whether the box sizes scale according to a power law or linearly. If the box sizes follow a power law, as shown in Figure 7.2 (A), then the local slopes are computed over regions consisting of a fixed number of ε values. If the box sizes scale in a linear fashion, then things are a little bit more complicated. In order to fit regions of constant size on a log-log plot, we select box sizes from within the range $\varepsilon_1 \leq \varepsilon \leq \varepsilon_2$ such that $\varepsilon_2/\varepsilon_1$ is constant (or approximately constant). This way, in both cases the local slopes are computed over a range of ε values satisfying $\varepsilon_2/\varepsilon_1 \approx m$. Throughout this thesis we will use the value $m = 1.5$ when computing the local slopes from the box-counting and generalized sandbox methods. We also use the convention of plotting the local slopes against the upper box size of the range, ε_2, as this provides a clear visualization of the scale at which a transition in the scaling relationship first occurs.

7.3.2 Some Simple Examples

In order to explicitly verify the results of Section 7.2.1 we have constructed a simple image in which a known fraction of pixels undergo a transition in the scaling relationship at a given scale, ε^*. Such an image, comprised of three unevenly spaced line segments, is shown in Figure 7.3 (A). As can be seen from the magnified illustration in Figure 7.3 (B), when $\varepsilon > \varepsilon^* = 64$ the mass-scaling relationship in the neighbourhood of the top two lines ($T = 2/3$ of the total pixels) transitions from $M(\varepsilon) = \varepsilon$ to $M(\varepsilon) = 2\varepsilon$. This corresponds to setting $T = 2/3$, $A = 1$ and $B = 2$ in Equations (7.4) and (7.5).

In Figures 7.4 (A) and (B), the local slopes (Δ_2 and Δ_0) between two consecutive box sizes are compared directly with Equations (7.4) and (7.5), respectively. To compute the local slopes, a minimum box size of three pixels was used, and the maximum box size was set to 50% of the image size. Box sizes were chosen according to a power law so that $\varepsilon_2/\varepsilon_1 \approx 1.5$ for each region. We observe excellent agreement between the theoretical results and the computational results from the generalized sandbox method at smaller box sizes. As the box sizes are increased, we begin to see the computational results deviate from the theory due to the edge effects which result from the use of computational methods.

Additionally, in Figure 7.4 (C), the local slopes from the box-counting method are presented. As expected, the behaviour of the local slopes is similar to that which results

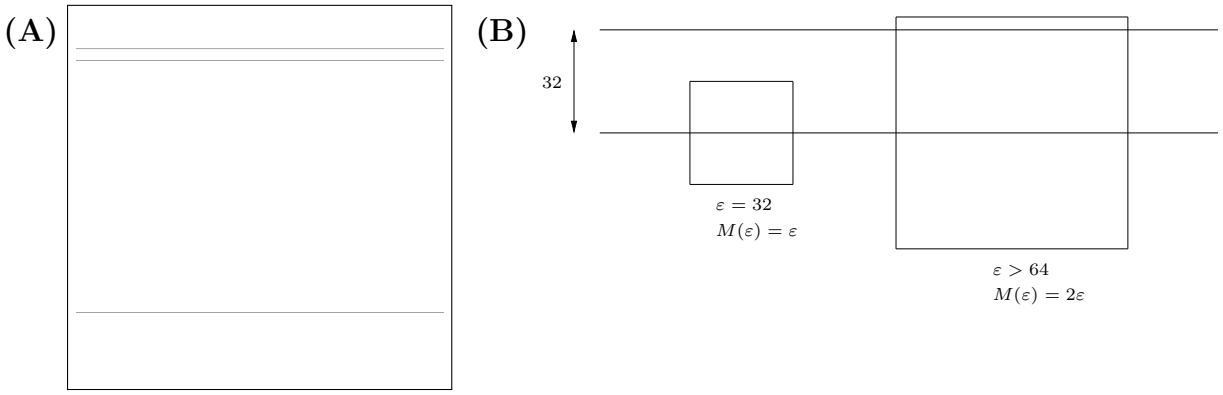

Figure 7.3: A sample image constructed to verify the results of Section 7.2.1. **(A)** The image contains three non-uniformly spaced lines. The top two lines are spaced 32 pixels apart and the bottom two lines are spaced 700 pixels apart. **(B)** Since the generalized sandbox method requires that boxes are centred on individual pixels, the transition from $M(\varepsilon) = \varepsilon$ to $M(\varepsilon) = 2\varepsilon$ occurs when $\varepsilon > 64$.

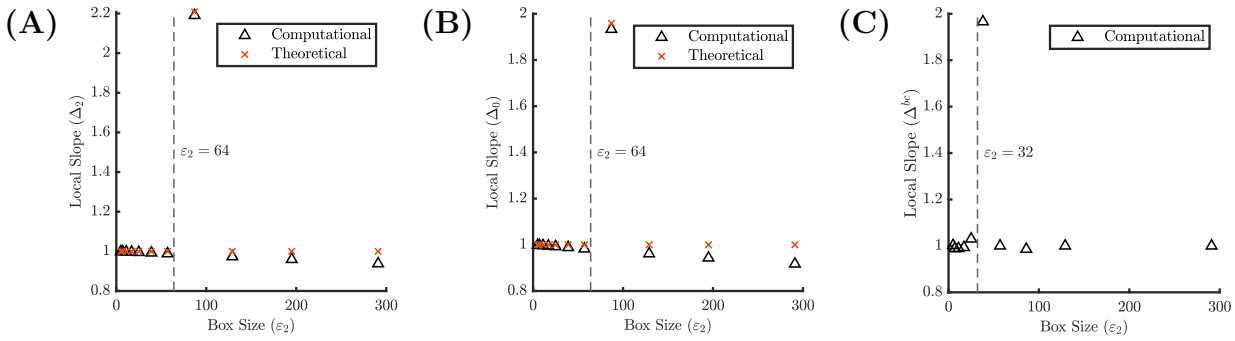

Figure 7.4: Estimating the fractal dimensions of Figure 7.3 (A) using the generalized sandbox and box-counting methods. **(A)** The local slopes (Δ_2) between ε_1 and ε_2, plotted as a function of the upper box size ε_2. The computational result is compared with the theoretical result from Equation (7.4). **(B)** The local slopes (Δ_0) compared with the theoretical result from Equation (7.5). **(C)** The local slopes (Δ^{bc}) computed using the box-counting method.

from the generalized sandbox method, however we note that the spike in the local slopes occurs earlier, at $\varepsilon_2 = 32$ instead of 64. Unlike the sandbox method, the box-counting method does not require boxes to be centred on individual pixels, and as such, the transition from a single line contained in a box to two lines within the same box occurs at a lower ε

value.

Figure 7.5 shows the corresponding log-log plots of $M(\varepsilon)$, $M^{-1}(\varepsilon)$ and $N(\varepsilon)$, respectively, also computed for the image in Figure 7.3. As is usually done, the line of best fit has been computed using linear regression on each data set, and the resulting slope (estimate of the dimension) and corresponding r^2 value are presented. We note that the r^2 values are

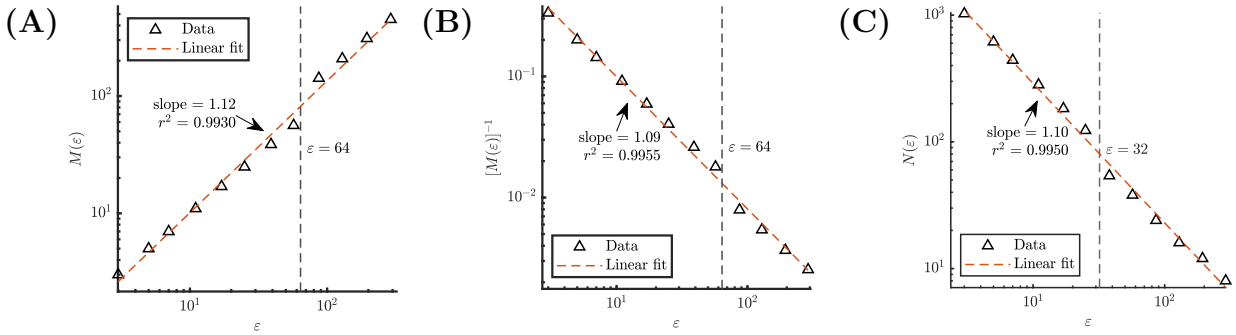

Figure 7.5: Estimating the fractal dimensions of Figure 7.3 (A) using the generalized sandbox and box-counting methods. **(A)** $M(\varepsilon)$ v.s. ε. **(B)** $M^{-1}(\varepsilon)$ v.s. ε. **(C)** $N(\varepsilon)$ v.s. ε.

close to one, which is usually interpreted as an indication of a good fit, however as we have discussed this is not a good metric for box-counting data. For example all three data sets presented here look reasonably linear at first glance, however if we take a closer look near the critical value of ε, we can clearly see a large deviation in the slope. This local spike biases the estimate of the dimension upwards, despite the image in question clearly being one-dimensional (a finite union of one-dimensional lines). If one simply used the log-log plots to estimate the dimension of this set, as is usually done, without considering the context added by the local slopes, the obvious conclusion would be that it is a fractal with a dimension (box-counting and correlation) of approximately $1.09 - 1.12$.

Although the results in Figure 7.4 verify our theoretical results quite nicely, the sample image we used is simple and not necessarily representative of what we expect to see from more complex branching structures. We now turn our attention to two additional sample images, shown in Figure 7.6, for which the theoretical results would be much more difficult to write down explicitly. However, the computational results illustrate the behaviour of the local slopes of the main components present in branching structures. The pair of angled lines in Figure 7.6 (A) are similar to a pair of branches emanating from a trunk in a tree-like structure, and the uniform grid in Figure 7.6 (B) resembles a more structured capillary network, not unlike Figure 7.1 (B). In Figure 7.7 we present the local slopes resulting from the box-counting and generalized sandbox methods applied to these two images.

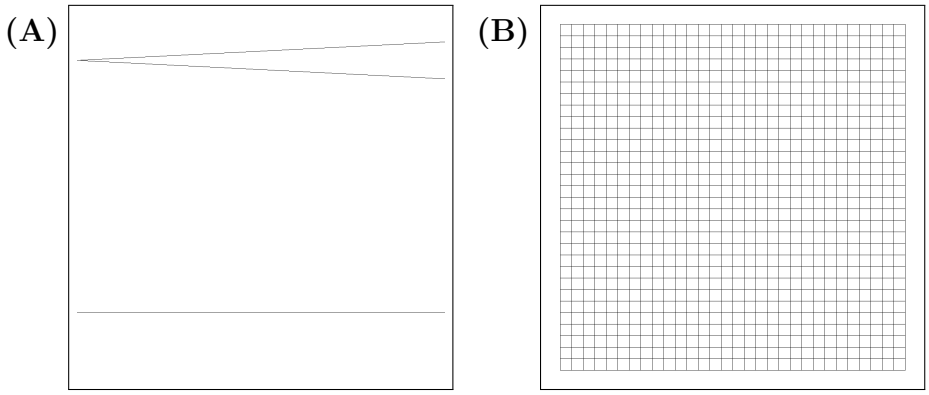

Figure 7.6: Two additional sample images generated to illustrate the main components of a general branching structure. In **(A)** a modified version of Figure 7.3 (A) with the top two lines at an angle of 0.1 radians from each other, and in **(B)** a simple grid with a uniform line spacing of 32 pixels.

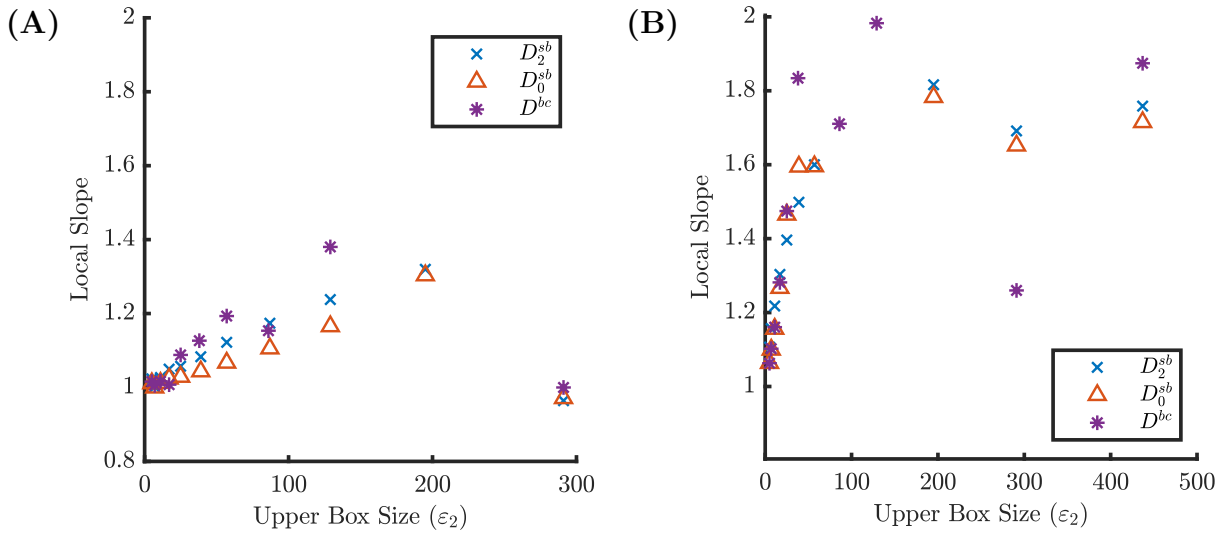

Figure 7.7: Plots showing the local slopes as a function of ε_2 for both images in Figure 7.6. Slopes are computed using the generalized sandbox (D_2^{sb} and D_0^{sb}) and box-counting methods (D^{bc}). **(A)** Local slopes for Figure 7.6 (A) - estimates of dimension yield $D_2^{sb} = 1.1034$, $D_0^{sb} = 1.0681$ and $D^{bc} = 1.0844$. **(B)** Local slopes for Figure 7.6 (B) - estimates of dimension yield $D_2^{sb} = 1.6484$, $D_0^{sb} = 1.6471$ and $D^{bc} = 1.6453$.

In Figure 7.7 (A) we see that when the two lines are placed at an angle, the local slopes

gradually increase, as opposed to the single spike which we saw when they were placed at a fixed distance from each other, i.e. parallel to each other. Instead of a single large transition in the proportionality constant, we have a series of smaller transitions occurring at increasingly large scales. Eventually we see the local slopes decrease back towards one when the box sizes are large enough that edge effects become relevant. This type of behaviour is akin to what we expect to observe in a more complex branching structure or fractal tree, but with many sets of branches of different sizes/scales, each branch causing increases in the average proportionality constant as we increase the scale.

In Figure 7.7 (B) we have the local slopes for a uniform grid with a line spacing of 32 pixels. A grid can be thought of as a special type of branching structure, in which there is no scaling (i.e. the parameter r is exactly 1) and the branches touch at a single point. Due to the lack of scaling, this set is not fractal (even if we consider an infinite number of branches), it will always be one-dimensional. At small scales (i.e. in the limit as ε goes to zero) we see this reflected in the local slopes, however as we look at scales larger than the line spacing of the grid, the local slopes rapidly increase due to transitions in the proportionality constant, eventually settling near a slope of two. At sufficiently large scales, a grid appears to fill space in the same way as a two-dimensional object, which is consistent with the measurements on capillary networks in both [85] and [68]. We note that the local slopes appear to approach the value two and then decrease as box sizes are increased further; this is due to the ever-present edge effects resulting from the use of discrete methods. Although it would not be correct to look at this plot and state that the fractal dimension of a grid (or grid-like structure) is two, the local slopes do provide us with some relevant information. They tell us that beyond some characteristic scale (in this case $\varepsilon^* = 32$) the structure is space-filling (i.e. effectively two dimensional), not fractal, which, as J.D. Murray writes,

> *... is a much more down to earth motivation if a branching structure is trying to maximise such things as spatial coverage without redundancy.* [80] (p. 369).

In order to more clearly visualize the effect that line spacing (or density) has on the results of estimating the fractal dimensions of grid-like structures, we have computed the local slopes for a number of images of grids, each with a different uniform line spacing. We have used the box-counting method with boxes sizes ranging from one pixel to 10% of the image size incremented linearly by two pixels. The local slopes are still computed over regions which satisfy $\varepsilon_2/\varepsilon_1 \approx 1.5$, however we have used more box sizes in order to capture the changes in the scaling behaviour in more detail. Figure 7.8 (A) shows the local slopes for five different grids with line spacing, n, ranging from $n = 4$ all the way

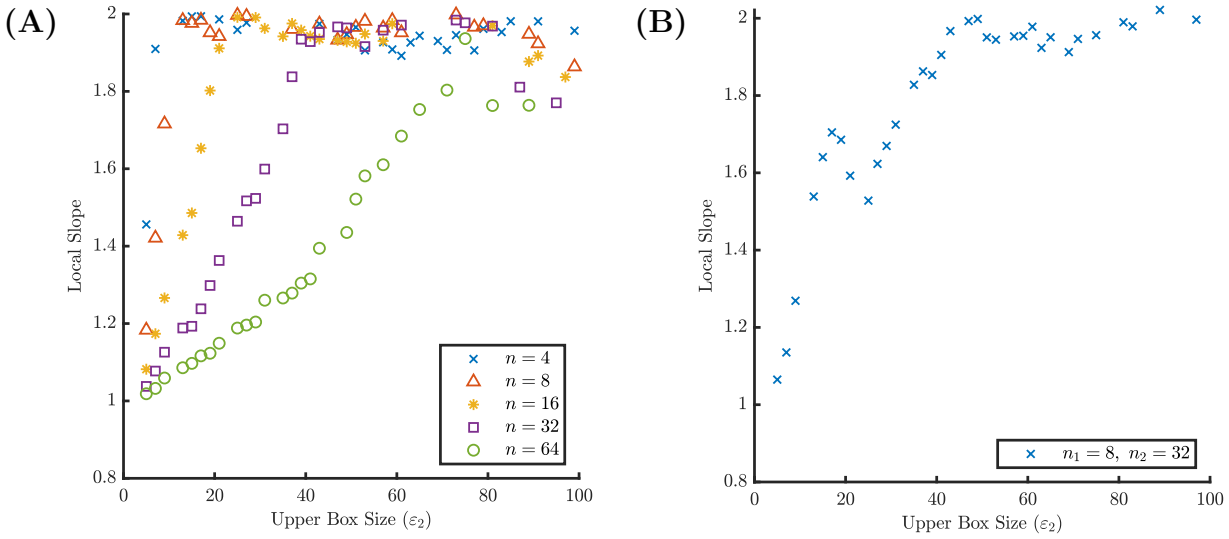

Figure 7.8: Plots showing the local slopes resulting from the box-counting method as a function of the upper box size (ε_2) for grids of varying line densities. In **(A)** we have the local slopes for five different images of grids with line spacing ranging from $n = 4$ to $n = 64$. In **(B)** we have the local slopes for a single grid with two different alternating line spacings, $n_1 = 8$ and $n_2 = 32$.

to $n = 64$. From the plot in (A) we can see that for each grid the local slopes undergo the same transition from one-dimensional to approximately two-dimensional, however the location of the transition is dependent on the line spacing of the grid. Although there is some smoothing due to computing the local slopes over a range of box sizes, the location of the transition corresponds roughly with the line spacing of each grid. When dealing with grid-like structures, we might call this size the characteristic scale of the structure, or in certain applications the characteristic pore size [68]. Of course, in more complex branching structures, there will be multiple scales (or pore sizes) present, each of which will have an effect on the local slopes.

We note here that similar results to those shown in Figure 7.8 (A) are obtained using the generalized sandbox method, however more variation in the local slopes is observed. This is due to the fact that the generalized sandbox method requires that the boxes are centred on each pixel. The different distances between each pixel and the corners of the grid cause transitions in the average mass-scaling behaviour to occur at different scales leading up to twice the characteristic spacing of the grid - we recall here the difference between the locations of transitions in the local slopes between the box-counting and generalized

sandbox methods as shown in Figure 7.4. This results in the local slopes looking "wavy" at smaller scales. At first, this may seem to contradict our previous comments regarding the robustness of the generalized sandbox method compared to the box-counting method, however those comments were made with regard to the estimation of a consistent scaling relationship. In this case, the scaling behaviour is non-constant, and the variation observed in the local slopes is not a flaw of the method, it is just a difference between how transitions in the scaling behaviour affect the local slopes resulting from both methods.

In Figure 7.8 (B) we show the local slopes resulting from the box-counting method applied to an image of a grid constructed with two distinct line spacings, $n_1 = 8$ and $n_2 = 32$. These two line spacings are implemented in an alternating manner so that both characteristic scales are represented equally in the image. We see two spikes in the local slopes for this image, one around each characteristic scale. Only when the box sizes used are larger than the largest characteristic scale present in the image, does the image appear space filling, or approximately two-dimensional. With only two characteristic sizes (which are not that close to each other), we are able to identify both spikes in the local slopes plot. However when dealing with more complex branching structures there may be many characteristic scales present, so the spike(s) in the local slopes may not always be so clear. Instead, many spikes may blend together, giving the appearance of a more gradual increase in the local slopes over many scales.

7.3.3 STARE Retinal Vasculature Images

In the previous section, we examined some simple examples to illustrate how changes in the scaling relationship can lead to increases in the local slopes, and consequently misleading results when attempting to estimate fractal dimensions from finite images. In this section, we look at how this effect presents itself when dealing with images of natural branching structures. As previously discussed, fractal dimension estimation methods have been widely applied to images of vascular networks, including healthy arteriovenous networks, tumour vasculature and both healthy and pathological retinal vasculature. In this section we will examine a set of images from he STructured Analysis of the REtina (STARE) Project [99]; a publicly available dataset consisting of approximately 400 raw color retinal images. A number of works use the box-counting and generalized sandbox methods to estimate the supposed fractal dimensions of these images [94, 92, 97]. Typically, healthy retinal vasculature has been found to have a dimension of around 1.71, the same dimension as DLA clusters, which has led to the widespread belief that retinal vasculature may form via the same process [94]. There are two problems with this assumption:

1. Retinal vasculature is a branching structure, and as such, violates the self-similarity assumption of the box-counting and generalized sandbox methods. This means that estimates of dimension made using these methods are flawed and may not relate to any potential fractal properties of the structure.

2. It is possible for two very different objects to have the exact same (or similar) fractal dimensions. For instance, the von Koch curve and Cantor dust, shown in Figures 5.2 and 5.3 (B), respectively, have the same dimension, yet it would be absurd to suggest that these two sets are generated by the same process.

Within the STARE database there is a standard subset of twenty images which have been manually segmented by two experts, Adam Hoover (AH) and Valentina Kouzentsova (VK) [100]. It is this subset of images which have been analysed in the aforementioned literature, and which we will consider in this work. Ten of the twenty standard images represent healthy retinal vasculature, and ten represent retinal vasculature with pathological manifestations. Samples of a typical normal image and a typical pathological image, segmented by the two observers AH and VK, respectively, are shown in Figure 7.9. We see that these images appear to be roughly space-filling and therefore we expect that, similar to the grids seen in the previous section, they will scale approximately as two-dimensional objects at large scales. We note that the images segmented by observer VK contain a much higher level of detail, as can be seen in Figure 7.9, and consequently have a higher average vessel density. This observation is consistent throughout the entire subset of the data and noted in much of the literature pertaining to the STARE data set.

The images in Figure 7.9 show the vascular structure very clearly, however they are not skeletonized. In order to extract just the structural information from the images we have skeletonized them prior to working with them, as is done in much of the literature. We have used the built-in `bwskel()` function in Matlab for skeletonization. In Figure 7.10 we present the results of applying the box-counting and generalized sandbox methods to the skeletonized version of Image 0162 segmented by observer AH. Linearly spaced box sizes with an increment of ten pixels between them were used to compute plots showing the local slopes and plots of $\overline{M}^{-1}(\varepsilon)$ vs. ε resulting from using the generalized sandbox method to estimate D_0. In plots (A) and (B) we have used a minimum box size of one pixel and a maximum box size of 50% of the image size in order to visualize the scaling behaviour of the image over a large range of scales. From (A) we can see that the local slopes start out near one when the box sizes are small - as expected since the image is comprised of line segments - and gradually increase to a value near two as the box sizes are increased. This is what we might expect based on our discussion in the previous section. A complex branching structure, like the retinal vasculature in Image 0162, is a collection of many individual

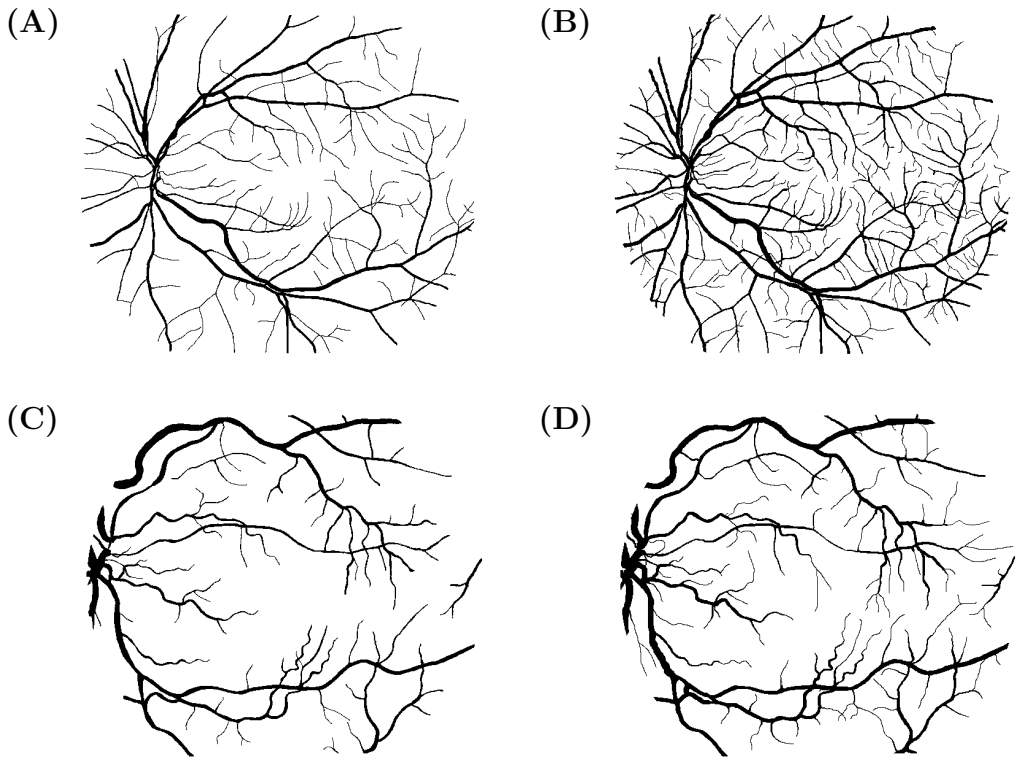

Figure 7.9: A typical healthy retinal vessel network (Image 0162) segmented by **(A)** Adam Hoover (AH) and **(B)** Valentina Kouzentsova (VK), and a typical pathological vessel network (Image 0001) segmented by **(C)** AH and **(D)** AK.

branches, like the pair of angled lines in Figure 7.6 (A), of varying lengths and angles. Each branch causes increases in the local slopes over some range of scales corresponding to the distances between the branches, along with the distances to neighbouring branches. All together, this creates the appearance of continuously increasing local slopes until the box sizes are large enough that the branching structure appears roughly space filling and the local slopes settle at a slope of nearly two. In (B) we can also see this behaviour on the plot of $\overline{M}^{-1}(\varepsilon)$ vs. ε fairly clearly.

Now, in Figures 7.10 (C) and (D) we have zoomed in on a smaller range of box sizes based on visual observation of the plot from (B), as is typically done in the literature when estimating fractal dimensions. In particular, plots (C) and (D) correspond to setting the minimum box size to 20 pixels and the maximum box size to 20% of the image size. We find that over this range of scales our estimate of the box-counting dimension D_0 is 1.64,

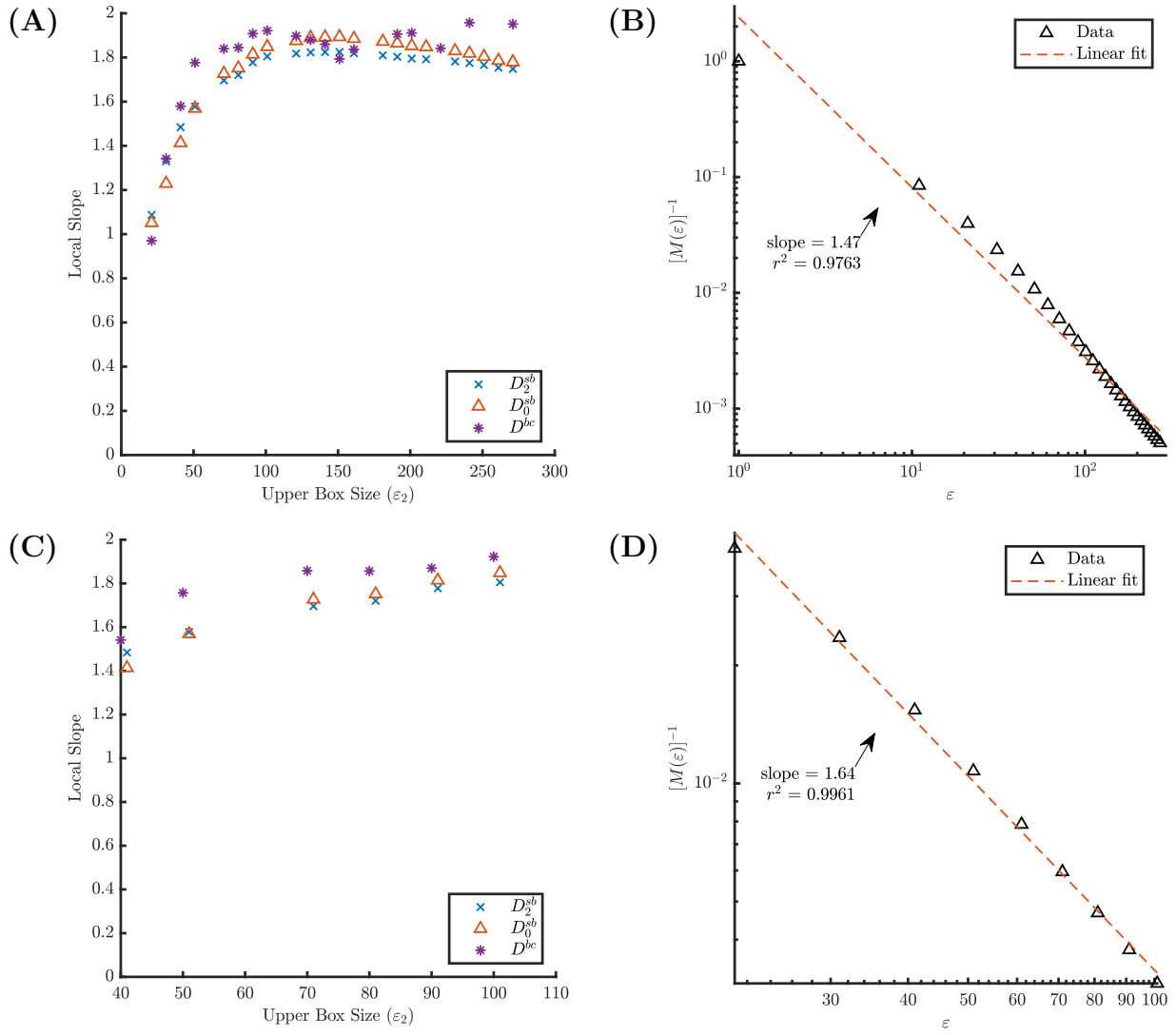

Figure 7.10: Estimation of the fractal dimension of Image 0162 segmented by AH (Figure 7.9 (A)). **(A)** Local slopes computed over box sizes from one pixel to 50% of the image size. **(B)** $\overline{M}^{-1}(\varepsilon)$ v.s. ε using box sizes from one pixel to 50% of the image size. **(C)** Local slopes computed over box sizes from one pixel to 50% of the image size. **(D)** $\overline{M}^{-1}(\varepsilon)$ v.s. ε using box sizes from 20 pixels to 20% of the image size.

which happens to be very close to the result obtained for the same image in [94]. In (D) we see that the data looks fairly linear over this range and the r^2 value is very high. Based

on the results in (D) alone, it would not seem unreasonable to state with confidence that the box-counting dimension of this vascular structures is 1.64, yet from our plot in (A) we know that the local slopes are rapidly increasing over this range of scales and the scaling behaviour is not constant.

As expected, we observe the same general behaviour in the local slopes computed for other images in the STARE data set, with changes only in the rate of increase in the local slopes. Just as in the previous section when we analysed grids of varying densities, we see that the rate at which the local slopes increase is related to the linear density of the image. As an example, we consider the version of Image 0162 which was segmented by observer VK. This image depicts the same branching structure, however, as was noted above, the images segmented by VK contain more fine detail (i.e. more small vessels and therefore more small gaps between vessels). Figure 7.11 shows the results of applying the box-counting and generalized sandbox methods to the skeletonized version of this image. In plots (A) and (B) we again have the local slopes and the corresponding log-log plot of $\overline{M}^{-1}(\varepsilon)$ vs. ε over the range of box sizes from one pixel to 50% of the image size. Here, we can clearly see the transition of the structure from one-dimensional at low scales to space-filling at larger scales, and we also note that this transition occurs earlier than it did in Figure 7.10.

In plots (C) and (D) we have the log-log plots of $\overline{M}^{-1}(\varepsilon)$ vs. ε over two smaller ranges of box sizes. In (C) the box sizes range from 20 pixels to 20% of the image size, the same parameters which were used to generate Figure 7.10 (D), and in (D) they range from ten pixels to 15% of the image size. We see that both plots appear reasonably linear, and have high r^2 values, despite the fact that the local slopes plot in (A) shows us that the slopes are increasing rapidly over both ranges of box sizes. We also see that the slopes, or estimates of dimension, over these two regions are quite different. It is easy to see how, without the context added by the local slopes plot, one could end up with wildly different results for the dimension of the same image just by using a different range of box sizes. Unfortunately the box sizes used for fractal dimension estimation are rarely recorded in the literature, making it fairly meaningless to compare the results across papers. Even within the same paper, it is not always clear if the same box sizes were used across images in a data set.

Now, we draw our attention to the plot in (C). The resulting slope through the data is 1.84, much higher than the result over the same range of scales for the same image segmented by observer AH (shown in Figure 7.10 (D)). Although both images represent the same branching structure, the additional detail included by observer VK causes spikes in the local slopes to occur earlier, and thus when the same range of box sizes is used to estimate dimension, the result is higher. We see time and time again in the literature that branching structures with higher densities and more fine structure are found to have a

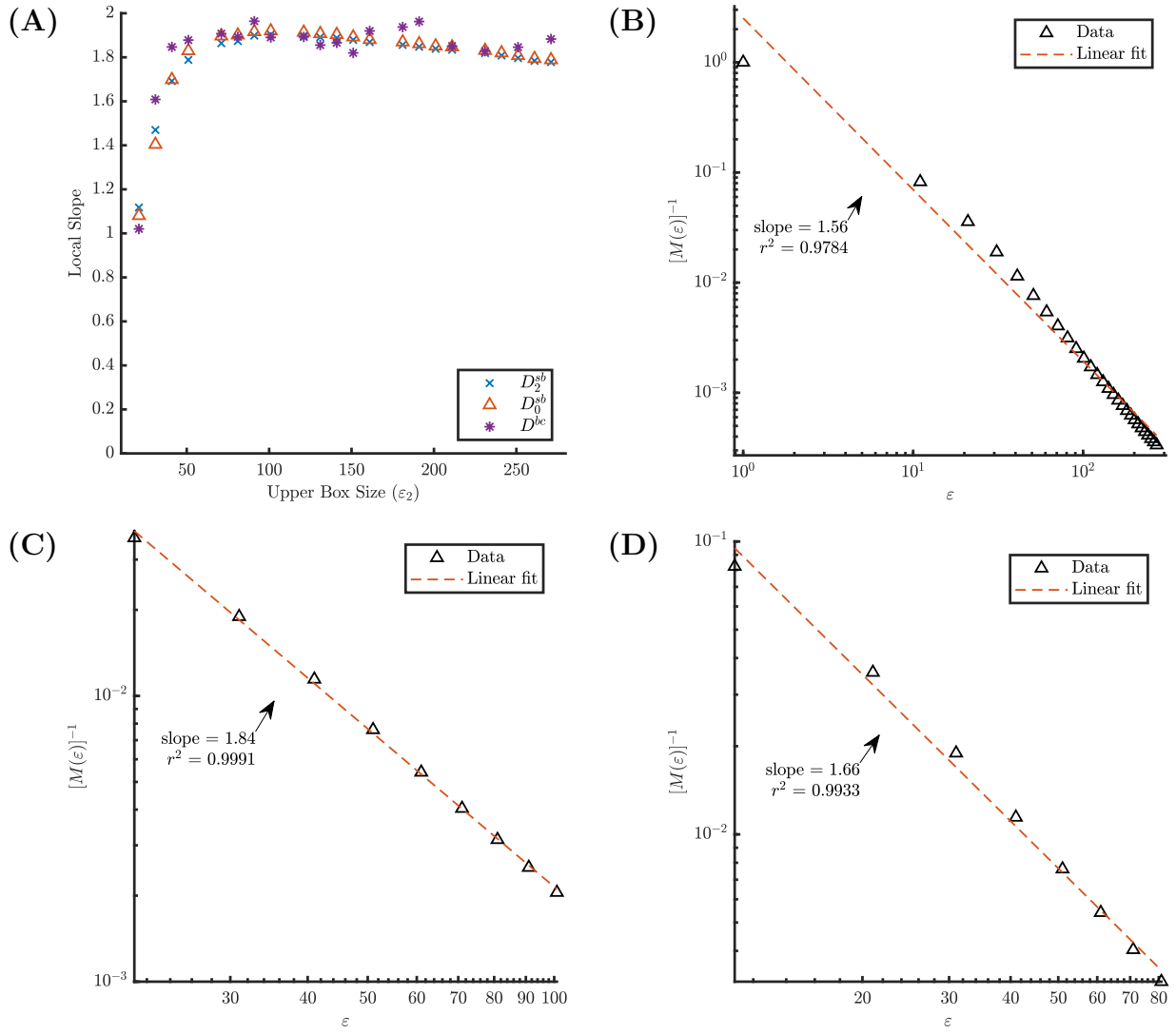

Figure 7.11: Estimation of the fractal dimension of Image 0162 segmented by VK (Figure 7.9 (B)). **(A)** Local slopes computed over box sizes from one pixel to 50% of the image size. **(B)** $\overline{M}^{-1}(\varepsilon)$ v.s. ε using box sizes from one pixel to 50% of the image size. **(C)** $\overline{M}^{-1}(\varepsilon)$ v.s. ε using box sizes from 20 pixels to 20% of the image size. **(D)** $\overline{M}^{-1}(\varepsilon)$ v.s. ε using box sizes from 10 pixels to 15% of the image size.

higher fractal dimension. In fact, in [94], which analyses the STARE data set, the images segmented by observer VK are noted to have a higher dimension than those segmented

by AH. This, despite the fact that both images are of the same underlying structure, the image segmented by VK simply contains one or two more iterations of branching. If an image is truly a finite approximation of an underlying fractal set - why should adding more detail change its fractal dimension?

The effect of density and characteristic pore sizes explains much of the success that has been found using fractal dimension estimation to classify branching structures found in nature. Images of pathological retinal vasculature, for instance, are typically found to have a lower dimension than healthy vasculature, and it turns out that pathological retinal vasculature is less dense than healthy retinal vasculature [97]. Another example is the classification of healthy and tumour vasculature networks using estimates of fractal dimension. As discussed in Section 7.1, tumour vasculature is consistently found to have a higher dimension than healthy vasculature, and as shown in Figure 7.1, tumour vasculature generally has a much higher vessel density than healthy vasculature. *It would seem that classification of such branching structures using estimates of fractal dimension has been successful in the past, not because of any fractal properties of the structures, but because the methods used are indirectly measuring vessel density.*

7.3.4 Computer Generated Fractal Trees

In this section we will show via computational examples that, as expected, the box-counting and generalized sandbox methods do not yield accurate estimates of the fractal dimension of computer generated fractal trees. Since it is not possible to know the theoretical dimension of real images of natural branching structures, such as the retinal vasculature images of the previous section, the results in this section provide the most compelling evidence against using these methods to estimate the fractal dimensions of branching structures in general.

We have implemented an iterative branching process in Matlab in order to generate finite approximations of fractal trees, specifically binary fractal trees (i.e. $N = 2$). Two such images are shown in Figure 7.12, each containing nine iterations of branching. We see that these trees correspond to the special case of $\theta_1 = \theta_2 = \theta$ and $r_1 = r_2 = r$ of the branching generator shown in Figure 5.5. For $r \geq 0.5$, and small enough such that the tree is non-overlapping, the theoretical dimension (Hausdorff and box-counting) of each tree is given by

$$D = \frac{\log(N)}{\log(1/r)} = \frac{\log(2)}{\log(1/r)}. \tag{7.6}$$

A derivation of this result for the box-counting dimension is presented in Appendix A. We note here that when $r = 0.5$ the dimension of the tree is one (by any definition of

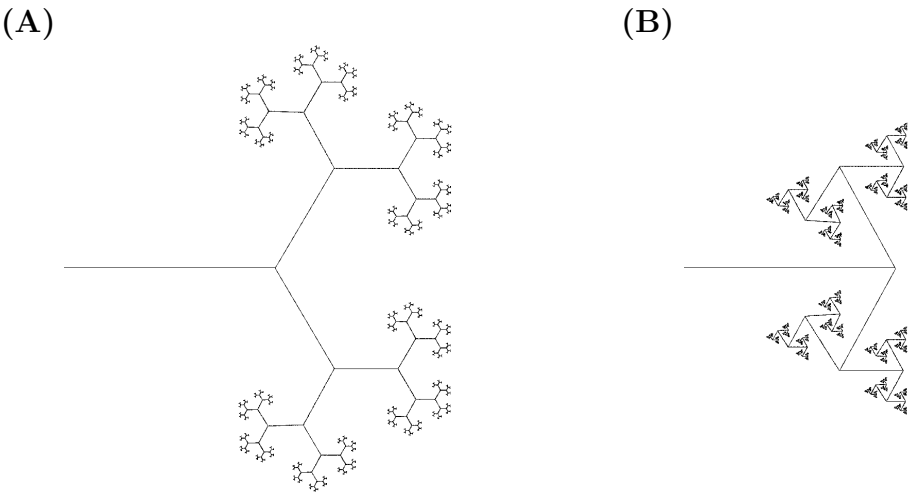

Figure 7.12: Two images depicting finite approximations of binary fractal trees with **(A)** $r = 0.55$, $\theta = \frac{\pi}{3}$, and **(B)** $r = 0.55$, $\theta = \frac{2\pi}{3}$.

dimension), however, despite having an integer dimension, we will still refer to this as a fractal tree as it is a limiting case. In Figure 7.12 we have shown two trees with the same branching length ratio, $r = 0.55$, however in (A) the branching angle is $\pi/3$ and in (B) it is $2\pi/3$. These two trees are visually quite distinct, with the tree in (A) expanding to fill a large two-dimensional area, while the tree in (B) folds in on itself and as a result occupies a smaller area and has a higher line density. Despite this, these two trees have the same "fractal dimension" - as we see in Equation 7.6, the dimension does not depend on the branching angle.

In Figure 7.13 we present plots showing the local slopes resulting from the box-counting and generalized sandbox methods for four distinct approximations of binary fractal trees. To generate these plots we applied both methods to images of simulated trees similar to those shown in Figure 7.12, however, in order to avoid artefacts in the images near the branch tips the initial trunk length was chosen such that the length of the outermost branches would be greater than a single pixel. As a result, each tree used approximately fills a 8192×8192 image. In computing the local slopes we set the minimum box size to be one pixel and the maximum at 50% of the image size. The box sizes were once again incremented in a linear manner by fifty pixels in order to sufficiently capture the changes in the scaling behaviour over the entire range of scales. The plots are cropped to show box sizes up to only 1000 pixels so that the scaling behaviour can be compared across the

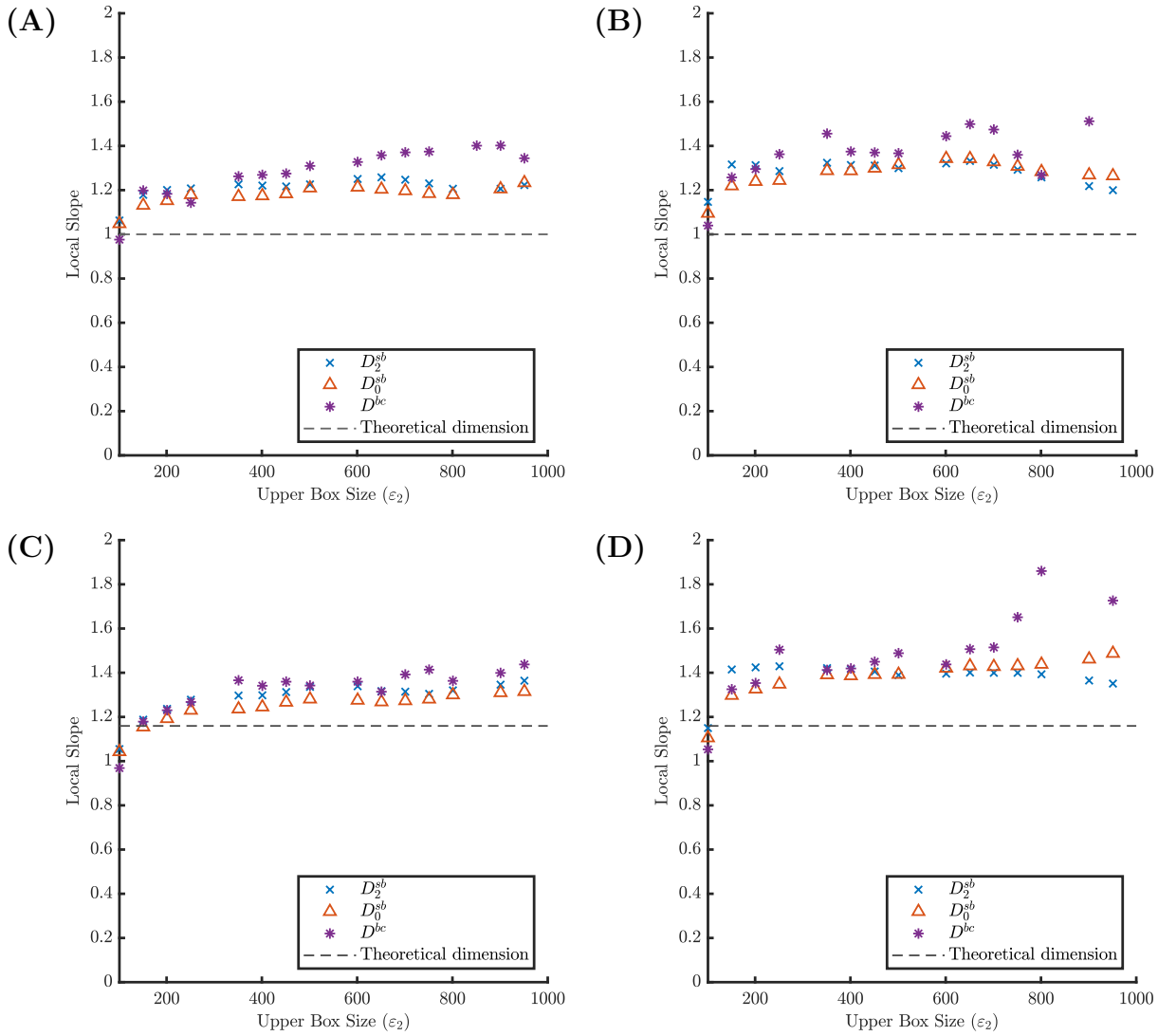

Figure 7.13: Plots showing the local slopes as a function of the upper box size ε_2 for four distinct fractal trees with the following branching parameters: **(A)** $r = 0.5$, $\theta = \frac{\pi}{3}$, **(B)** $r = 0.5$, $\theta = \frac{2\pi}{3}$, **(C)** $r = 0.55$, $\theta = \frac{\pi}{3}$, and **(D)** $r = 0.55$, $\theta = \frac{2\pi}{3}$.

same range of scales for each tree. In plots (A) and (B) we have the local slopes for trees with the same length ratio, $r = 0.5$, and different branching angles, $\theta = \pi/3$ and $\theta = 2\pi/3$, respectively. In (C) and (D) the length ratio is increased to $r = 0.55$ and again we have $\theta = \pi/3$ and $\theta = 2\pi/3$. Consequently the dimension of the trees in (A) and (B) is $D = 1$,

while the dimension of the trees in (C) and (D) is $D \approx 1.1594$. In each plot this theoretical dimension is indicated by a horizontal dashed line.

Since the trees are composed of finite line segments, at small scales we see that the local slopes are approximately equal to one, as expected. Although the local slopes do pass through the true dimension of each tree briefly, we see that the slopes continue to increase well beyond this point in every case due to the transitions in the scaling relationship described in Section 7.2.1. We notice that, despite the trees having the same theoretical dimension, the local slopes in (B) increase more rapidly than those in (A), and similarly for (D) compared to (C). This is a result of the trees with the larger branching angle, $\theta = {}^{2\pi}/_{3}$, having a higher line density and thus smaller gaps between branches, as was shown in Figure 7.12. Although the theoretical dimension is unaffected by the change in branching angle, the local slopes clearly show us the difference in density between these visually distinct trees. At some point we see that the edge effects start to compete with the increases in slope causing somewhat of a "false plateau" at some fractional value between one and two. If we were to take this plateau as evidence of a consistent scaling relationship and estimate the fractal dimension from these plots we see that we would significantly overestimate the dimension of these trees. We note here that these trees are less space-filling than the vasculature images from the previous section, and as such we do not see two-dimensional behaviour, even at fairly large scales. As we can see in Figure 7.12, some regions of the tree (namely the trunk) appear one-dimensional even at very large scales.

Chapter 8

Characterizing Naturally Occurring Branching Structures: The Way Forward

The results of the previous section leave us with some obvious questions. How do we reconcile our results with our knowledge that fractal trees are indeed fractal? One might wonder if it is even possible to measure the theoretical dimension from finite image of a fractal tree directly, and if so, how? The scaling properties of the tree are surely encoded in the image in some way. At this point, we recall that a fractal tree can be thought of as the union of a one-dimensional object (the trunk and branches) and a D-dimensional object (the tree canopy, or 'leaves'). Although the entire tree is not strictly self-similar, it is easy to show that the canopy, which is similar to a Cantor dust (see Figure 5.3 (B)), is self-similar. As a result, we might expect our usual methods of estimating the dimension to be successful if applied directly to finite approximations of the tree canopy.

8.1 Estimating Fractal Dimensions From Tree Canopies

In order to construct finite approximations of the canopies corresponding to the four binary fractal trees discussed in the previous section, we simply extract the endpoints of the upper branches from the image of the tree itself. This can be done very simply using the `bwmorph()` function in Matlab to extract all of the endpoints and removing the single

endpoint corresponding to the trunk. Each image of a tree canopy is comprised of exactly 512 branch tips, represented by a single pixel (or point) in the image. Figure 8.1 shows the local slopes plots resulting from the box-counting and generalized sandbox methods applied to each fractal tree canopy. As before, we used box sizes ranging from one pixel

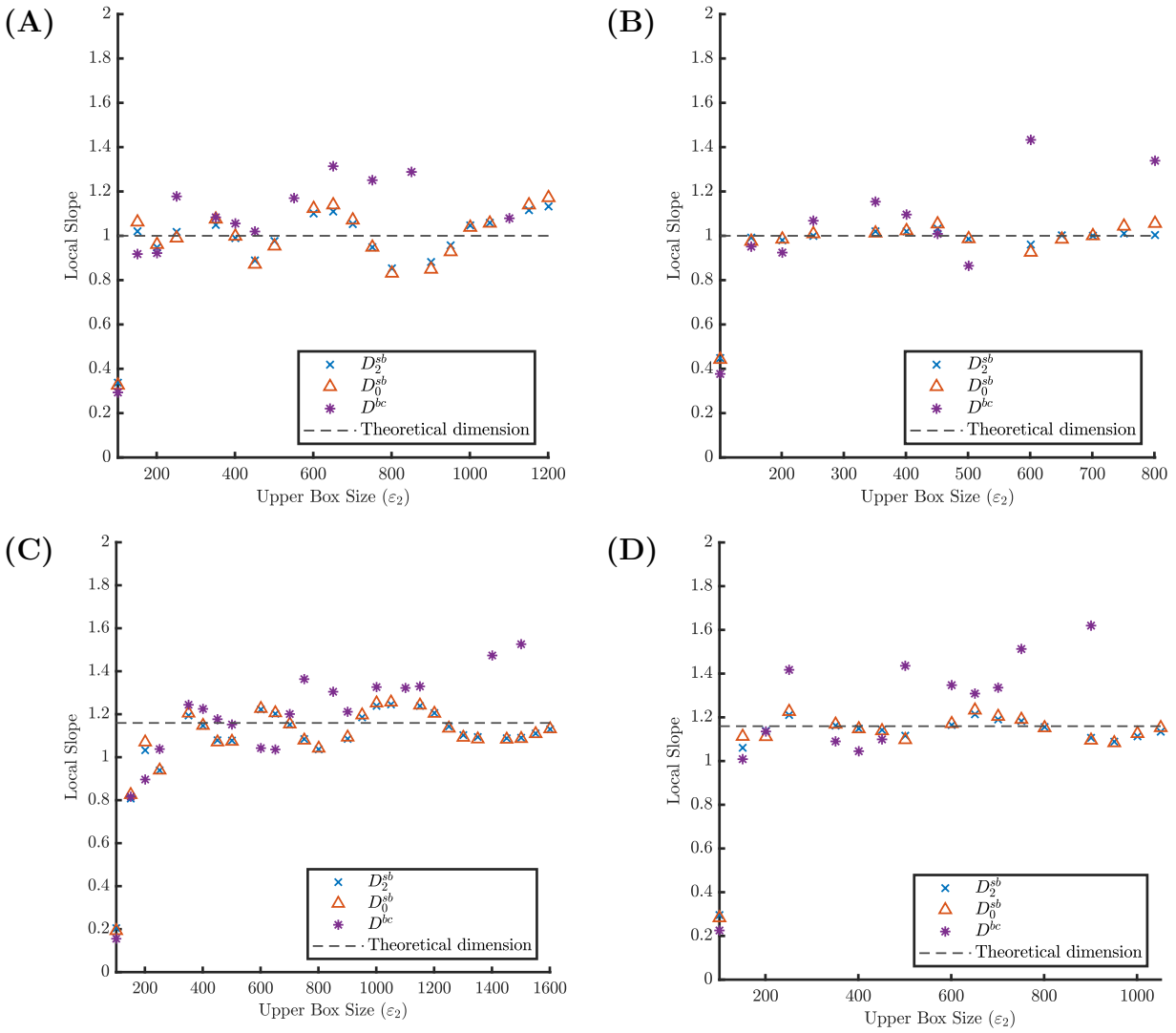

Figure 8.1: Plots showing the local slopes resulting from the box-counting and generalized sandbox methods as a function of the upper box size ε_2 for four distinct fractal tree canopies. **(A)** $r = 0.5$, $\theta = \pi/3$. **(B)** $r = 0.5$, $\theta = 2\pi/3$. **(C)** $r = 0.55$, $\theta = \pi/3$. **(D)** $r = 0.55$, $\theta = 2\pi/3$.

to 50% of image size, incremented linearly by fifty pixels at a time. The local slopes were computed over a range of box sizes satisfying $\varepsilon_2/\varepsilon_1 \approx 1.5$, as usual. At small scales, the canopies appear roughly zero-dimensional, which is to be expected as in the finite case the canopy is a finite collection of points. As we increase the box sizes, we see that the local slopes increase and hover around the true box-counting dimension of the canopy (and thus the tree itself). We note that the canopy images are quite sparse, so we see dips and peaks in the local slopes when the change in box sizes is small compared to the scales present in the image. In general, there is a fine balance to strike between using many box sizes (resulting in more variability in the local slopes) and using too few box sizes (and thus not being able to clearly see if/where there is a plateau in the local slopes).

The examples in Figure 8.1 illustrate how we can use the box-counting and generalized sandbox methods to estimate the fractal dimensions of perfectly self-similar canopies. However, more realistically, we expect the canopies of naturally occurring branching structures to be statistically self similar at best. Figure 8.2 shows the same results for the box-counting and generalized sandbox methods applied to two images depicting finite approximations of statistically self-similar tree canopies. We can can construct tree canopies

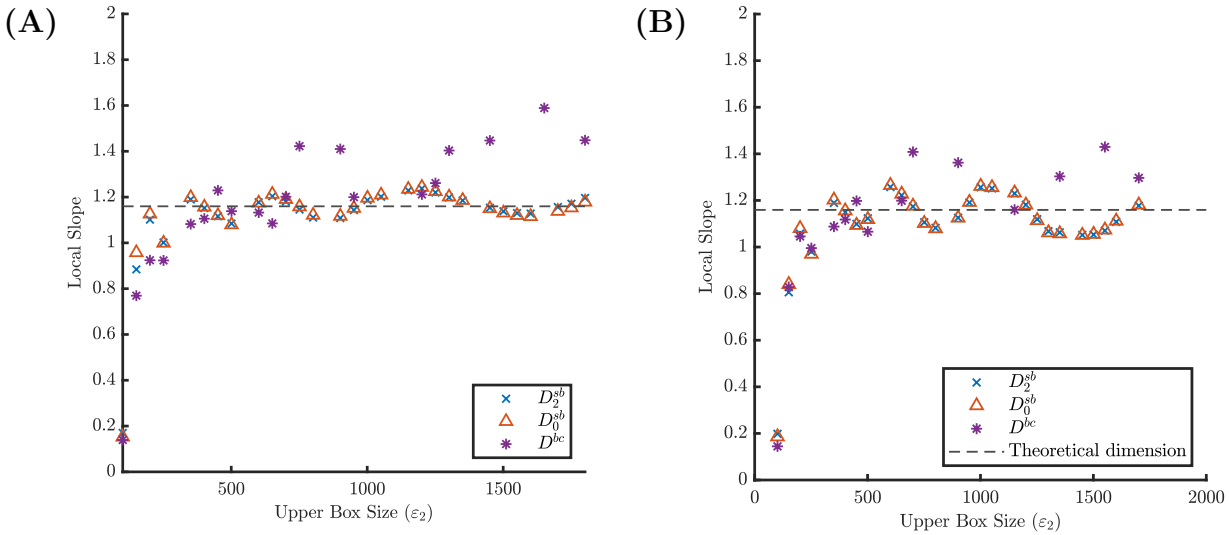

Figure 8.2: Plots showing the local slopes resulting from the box-counting and generalized sandbox methods as a function of the upper box size ε_2 for two statistically self-similar fractal tree canopies. In **(A)** $r \sim \mathcal{U}(0.51, 0.59)$, $\theta = \frac{\pi}{3}$ and in **(B)** $r = 0.55$, $\theta \sim \mathcal{U}(\pi/3 - 0.1, \pi/3 + 0.1)$.

with statistical self-similarity by introducing some randomness into the branching process.

In both cases we have fixed the bifurcation ratio as $N = 2$, and varied either the branching length ratio, r, or the branching angle, θ, randomly. We will use the notation C_A and C_B to denote the infinite sets which have been approximated by the finite canopies used to generate plots (A) and (B), respectively. To generate the canopy C_A, the value of r has been chosen from a uniform distribution on each iteration, such that the mean value is $\bar{r} = 0.55$, and θ has been fixed at a value of $\pi/3$. The canopy C_B was generated using a fixed value of $r = 0.55$ and θ chosen from a uniform distribution with mean value $\bar{\theta} = \pi/3$. The theoretical box-counting dimension for each (infinite) canopy is indicated by a horizontal line on the corresponding plot.

We point out here that when dealing with statistically self-similar sets we need to be careful when evaluating the theoretical dimensions. The theoretical dimensions of the two canopies, C_A and C_B, are close in value, but are not equal. When r is varied, as in C_A, we must treat r as a random variable in our derivation of the dimension. With r chosen from the uniform distribution described above, we can apply Theorem 15.2 from [51] to show that, with probability one, the box-counting (and Hausdorff) dimension of the canopy is s, where s is the solution to the expectation equation

$$\mathbb{E}\left[2r^s\right] = 1. \tag{8.1}$$

Solving this expression for $r \sim \mathcal{U}(0.51, 0.59)$ gives us $s \approx 1.1597$, i.e. the box-counting dimension of the set C_A is

$$dim_B(C_A) \approx 1.1597. \tag{8.2}$$

When r is fixed and θ is chosen randomly, things are a little bit simpler. In this case, as long as we choose values of θ such that the tree remains non-overlapping, we can use the same argument from Appendix A to show that

$$dim_B(C_B) = \frac{\log(2)}{\log(1/r)} \approx 1.1594. \tag{8.3}$$

As before, we observe that at large enough scales the local slopes hover nicely around the true box-counting dimension of the canopy, even when we only have statistical self-similarity. Based on the results in this section we might expect that if natural branching structures are in fact statistically self-similar, then we should be able to estimate their fractal dimensions by applying traditional fractal dimension estimation methods to their canopies, if those canopies are present in the image. We have seen that this method is applicable to "nice" simulations of fractal trees, however when attempting to extend this to natural branching structures we must keep the following points in mind:

1. In reality, branching structures are typically 3-dimensional and common practice is to produce 2-dimensional images by projecting a slice of the structure onto the 2-D plane. This causes overlap and disrupts the spatial organization of the canopy. To measure the dimension of a 3-dimensional canopy would require 3-D volumetric images, which typically are much more costly and time-consuming to generate.

2. Unless a very large number of generations of branching are imaged, the tree canopy will be a very sparse set. This leads to wavy behaviour in the local slopes, as observed in Figures 8.1 and 8.2, making it difficult to identify a plateau. Most real-life branching structures do not contain a particularly large number of generations, or if they do, it is at a scale too small to resolve with standard imaging techniques.

3. Related to the above point, edge effects will be particularly prevalent since every point of the canopy may be considered as an edge, once again making it difficult to identify a plateau accurately. As we have seen (Figure 7.12), edge effects can lead to false plateaus for non-homogeneous sets.

4. Finally, as was shown in [81], random distributions of points can appear fractal over certain ranges of scales. As such, one should only apply fractal dimension estimation methods if there is reasonable cause to assume (statistical) self-similarity.

Although we have seen that the box-counting and generalized sandbox methods can be used to successfully estimate the dimensions of simulated fractal tree canopies, it is unlikely that these methods will be successful when applied to images of natural branching structures. Perhaps one day, with further advances in the quality of imaging data, this may be a possibility, but for now other means of characterizing natural branching structures should be considered.

8.2 Direct Estimation of the Branching Parameters

If we continue with the assumption that the branching structures we see in nature do in fact have fractal properties, then we might consider directly estimating the branching parameters in order to estimate their fractal dimensions. We know from Equation (7.6) that (for certain values of the branching parameters) the box-counting (and Hausdorff) dimension of a fractal tree depends on both the bifurcation ratio, N, and the length ratio, r, of the branching generator. There is some research, pertaining to the study of fluvial landscapes (i.e. stream and river networks), on estimating these parameters directly from

images of branching structures [101, 77, 102]. In this field of study, fluvial landscapes are considered as tree-like structures and it is common to use a numbering system to order the branches in a hierarchical fashion. Most commonly used is the Strahler stream order, in which branches are ordered from the outside in such that all outermost branches are of order one and subsequent parent branches are ordered based on the order of their children. Using this ordering system, with u representing the branch order, the bifurcation ratio can be defined as

$$r_B = \frac{N_u}{N_{u+1}} \tag{8.4}$$

where N_u and N_{u+1} are the number of branches of order u and $u+1$, respectively. Similarly, the length ratio can be defined as

$$r_L = \frac{L_u}{L_{u-1}} \tag{8.5}$$

where L_u and L_{u-1} are the mean length of branches of order u and $u-1$, respectively.

Fluvial branching networks are typically assumed to follow Horton's Laws [102, 77] which state that the bifurcation and length ratios are constant within a given catchment (geographical area). If this is true then this implies strict self-similarity of the branching network and it is easy to show that [77]

$$dim_S = dim_H = dim_B = \frac{\log(N)}{\log(1/r)} = \frac{\log(r_B)}{\log(r_L)}. \tag{8.6}$$

For branching networks which are strictly self-similar, or at least approximately self-similar, direct computation of r_B and r_L from the data could be a reliable method with which to estimate their fractal dimensions. Typically, natural branching structures are not strictly self-similar, and so values of r_B and r_L are computed by averaging over the entire structure. Let us use the notation

$$D_{\bar{r}} = \frac{\log(\overline{r_B})}{\log(\overline{r_L})}, \tag{8.7}$$

where $\overline{r_B}$ and $\overline{r_L}$ are the average values of r_B and r_L, to denote estimates of the dimension computed in this way.

Now, from our discussion in the previous section we know that the estimate $D_{\bar{r}}$ in Equation (8.7) does not always correspond exactly to the fractal dimension of the corresponding branching structure. In the more realistic case of statistical self-similarity,

the theoretical dimension will be the solution to an expectation equation similar to the one in Equation (8.1). That being said, there is only a fairly small range of branching parameters which generate non-overlapping (i.e. physically realizable) trees. Therefore, for most realistic distributions, the value $D_{\bar{r}}$ should provide a good approximation of the theoretical dimension of branching structures - as we saw in the example from the previous section.

An added benefit of using the value $D_{\bar{r}}$ to classify branching structures is that, if we strip away the pretence that its value corresponds to a fractal dimension, we can use this value to characterize the complexity of a wide variety of branching structures - not just those with fractal properties. For instance, branching structures which satisfy $r_B \leq r_L$ can be shown to have finite length and thus a dimension of one. This means that fractal dimensions provide us with no way to distinguish between such branching structures, yet estimates of $D_{\bar{r}}$ will vary and thus allow for better classification of these sets. We may also gain further information by examining how $D_{\bar{r}}$ varies over different regions of the structure, or even more specifically how each of the branching parameters, r_B and r_L, varies over the structure.

We should make note here that the practicality of this method does, once again, rely on the quality and nature of the imaging data which one has available. Most branching structures which occur naturally are three dimensional and projected into two dimensions for imaging, which can result in significant overlap in the images. In order to accurately estimate the ratios r_B and r_L from an image of a branching structure we require minimal overlap of the branches in the two-dimensional projection, or three-dimensional data in order to avoid the issue of overlap entirely.

8.3 A Simpler Approach - Do We Even Need Fractal Methods?

Based on the above discussions, it is clear that estimating the fractal dimensions of naturally occurring branching structures is a much more nuanced and difficult problem than many previous works have suggested. Consequently, one may start to wonder if all of this is even worthwhile. Branching networks aside, how many natural structures are sufficiently self-similar over a large enough range of scales that we might have a chance to truly estimate their fractal dimensions and what do we stand to gain by doing so? It would seem that the question we really want to ask ourselves is: why are we estimating the fractal dimensions of naturally occurring objects in the first place? Once again, quoting J.D. Murray from [80],

a paper which asked similar questions,

> *A particular and widespread misconception about fractal theory arises because it can create objects which look remarkably like many natural structures such as trees, weeds, flowers, butterfly wing patterns and so on, and this is often taken to be a biological explanation of how these structures and patterns are formed. Although fractal-like patterns may be reasonable graphical representations of such natural shapes, they say essentially nothing about the biological processes and mechanisms which are involved in their development.* [80] (p. 369).

Indeed, this assessment has been echoed by others, including A. Bejan, the inventor of "constructal theory" - see, for example, [103], Section 1.2, "The Hardest Questions." As we have noted previously, it is possible (and quite common) for multiple very different objects to have the same (or very similar) fractal dimensions. Therefore, fractal dimension alone does not allow us to draw conclusions about the process by which a structure was generated or even necessarily its visual appearance (consider the appearance of the von Koch curve and Cantor dust as an example).

Going forward, what we can say for certain is that methods like the box-counting and generalized sandbox methods cannot be used to estimate fractal dimensions from finite images of branching structures. This is due to the non-homogeneity of branching structures in general. As such, the values resulting from these methods should not be referred to as the fractal dimension of the structure or used to make claims about the biological processes used to generate such networks. Fractal geometry certainly has its place, however it is important to distinguish between theoretical fractal sets and natural objects which may appear approximately fractal over finite ranges of scales. An understanding and respect for this difference is imperative when considering the application of fractal theory to real objects. Our thoughts on this matter are nicely summarized by one final quotation from Kenneth Falconer,

> *The distinction between 'natural fractals' and the mathematical 'fractal sets' that might be used to describe them was emphasised in Mandelbrot's original essay, but this distinction seems to have become somewhat blurred. There are no true fractals in nature. (There are no true straight lines or circles either!) If the mathematics of fractal geometry is to be really worthwhile, then it should be applicable to physical situations.* [51] (p. xxix).

With all that being said, the results from previous studies, using what we now know are flawed methods, can still provide us with some useful information on how to characterize

branching structures. For instance, a positive correlation has generally been observed between vessel density and estimates of the box-counting or sandbox dimension. There are many examples in the literature in which estimates of fractal dimensions were found to correlate well with some morphological property of branching structures, for example, distinguishing between healthy and tumour vasculature or pathological and normal retinal vasculature. Perhaps what this tells us is that, following the principle of Occam's Razor, we don't need a complicated fractal theory to distinguish between such branching structures. Instead, perhaps simpler measurements, such as average vessel density or local measures of density may be sufficient. Furthermore, as we have seen, analysis of the local slopes plots resulting from the box-counting and generalized sandbox methods can provide more context regarding the spatial density of images and the presence of characteristic scales. Figures 7.7 and 7.8 in this thesis, and the results of [68] provide a good example of this phenomenon, in which the characteristic pore size of a space-filling network is evident from the local slopes plot.

Chapter 9

Concluding Remarks

This thesis consists of two distinct parts, each of which is dedicated to the examination of a particular approach used to characterize a specific type of medical image. Part I focusses on the design of efficient image descriptors for digital histopathology. The examination of pre-existing methodology led to the proposal of two approaches to improve the image search performance of the projection-based ELP descriptor, a descriptor which was previously shown to perform well at retrieving similar histopathology images. First, a careful analysis of the ELP method revealed some weaknesses in the method, such as long histograms, redundancy, and a lack of rotational invariance. Based on this analysis, an improved frequency-based encoding was proposed to generate shorter and more meaningful histograms. Additionally, the use of stain-separated and RGB colour images as inputs to generate descriptors was investigated and found to lead to a significant improvement in image search performance over the greyscale images which were used in prior works. Improvements were observed using colour images to generate four different image descriptors, indicating that the benefits of using colour images as inputs extends beyond just the ELP descriptor, but could apply to any and all image descriptors which have been developed for digital histopathology, and to those which will be developed in the future.

In Part II of the thesis, a close look into the methods used to estimate fractal dimensions from images of natural structures revealed a significant flaw in these methods - the assumption of self-similarity which is required for these methods to work. Further analysis showed that branching structures, a class which encompasses many natural objects - including complex vascular networks - are inherently not self-similar due to the branching process by which they are formed. Although such structures may have a well-defined fractal dimension when their limiting behaviour is considered, they exhibit non-constant scaling behaviour over finite scales. Therefore, standard methods such as the box-counting

and sandbox methods were found to be an ineffective means of estimating their fractal dimensions. In fact, computational examples showed that estimates of the fractal dimensions of branching structures yielded by these methods are highly dependent on the scales which are used in the estimation. Some alternative means of characterizing naturally occurring branching structures were suggested.

Although the two methods described above are quite unrelated - apart from their use to characterize medical images - both were analysed with the goal of gaining a deeper understanding of the exact mechanisms which lead to their success (or failure as it turns out in Part II). It is the hope of the author that the lessons learned from this work will be used to improve future research efforts in both areas.

Letter of Copyright Permission for Figure 7.1

Alison Cheeseman

From: Katie Zimmerman [redacted]
Sent: December 15, 2021 4:35 PM
To: Alison Cheeseman; permissions-lib
Subject: RE: Permission to reuse a figure from MIT thesis

Hi Alison,

Happy to help with this. MIT theses fall under the MIT Libraries Permissions Policy, which automatically grants permission for academic reuse of figures. That's sufficient on it's own, but in case it's also helpful to have a specific statement:

In response to your December 15, 2021, inquiry, the copyright notice corresponding to the 1996 thesis "Fractal vasculature and vascular network growth modeling in normal and tumor tissue," by Yuval Gazit, attributes ownership to the Massachusetts Institute of Technology ("MIT"). Accordingly, MIT hereby grants permission to use Figure 4-2 (subfigures a), b) and c)) in your forthcoming article and PhD thesis, provided that any such use includes appropriate citation. This authorization to use the copyrighted material also applies to any future editions, revisions, or translations and to mechanical and electronic storage on any carrier.

Please let me know if you have any questions or need anything else.

Best,
Katie

Katie Zimmerman, JD, MLIS (pronouns: she/her)
Director, Copyright Strategy
MIT Libraries
[redacted]

References

[1] Alison K. Cheeseman, Hamid R. Tizhoosh, and Edward R. Vrscay. A compact representation of histopathology images using digital stain separation and frequency-based encoded local projections. In Fakhri Karray, Aurélio Campilho, and Alfred Yu, editors, *Image Analysis and Recognition*, pages 147–158, Waterloo, ON, 2019. Springer International Publishing.

[2] Alison K. Cheeseman, Hamid R. Tizhoosh, and Edward R. Vrscay. Studying the effect of digital stain separation of histopathology images on image search performance. In Fakhri Karray, Aurélio Campilho, and Alfred Yu, editors, *Image Analysis and Recognition*, pages 262–273, Póvoa de Varzim, Portugal, 2020. Springer International Publishing.

[3] Alison K. Cheeseman and Edward R. Vrscay. Estimating the fractal dimensions of vascular networks and other branching structures: Some words of caution. *Mathematics*, 10(5):1–21, 2022.

[4] Felix Ritter, Tobias Boskamp, André Homeyer, Hendrik Laue, Michael Schwier, Florian Link, and Heinz-Otto Peitgen. Medical image analysis. *IEEE Pulse*, 2(6):60–70, 2011.

[5] Angel Cruz-Roa, Ajay Basavanhally, Fabio Gonzalez, Hannah Gilmore, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, and Anant Madabhushi. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In Metin N. Gurcan and Anant Madabhushi, editors, *Medical Imaging 2014: Digital Pathology*, volume 9041. International Society for Optics and Photonics, SPIE, 2014.

[6] Abtin Riasatian, Morteza Babaie, Danial Maleki, Shivam Kalra, Mojtaba Valipour, Sobhan Hemati, Manit Zaveri, Amir Safarpoor, Sobhan Shafiei, Mehdi Afshari, Maral Rasoolijaberi, Milad Sikaroudi, Mohd Adnan, Sultaan Shah, Charles Choi, Savvas

Damaskinos, Clinton JV Campbell, Phedias Diamandis, Liron Pantanowitz, Hany Kashani, Ali Ghodsi, and H.R. Tizhoosh. Fine-tuning and training of densenet for histopathology image representation using TCGA diagnostic slides. *Medical Image Analysis*, 70:1–11, 2021.

[7] Metin N. Gurcan, Laura E. Boucheron, Ali Can, Anant Madabhushi, Nasir M. Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 2(2):147–171, 2009.

[8] Anant Madabhushi. Digital pathology image analysis: Opportunities and challenges. *Imaging in Medicine*, 1(1):7–10, 2009.

[9] Yibing Ma, Zhiguo Jiang, Haopeng Zhang, Fengying Xie, Yushan Zheng, Huaqiang Shi, Yu Zhao, and Jun Shi. Generating region proposals for histopathological whole slide image retrieval. *Computer Methods and Programs in Biomedicine*, 159:1–10, 2018.

[10] Liron Pantanowitz, Ashish Sharma, Alexis B. Carter, Tahsin Kurc, Alan Sussman, and Joel Saltz. Twenty years of digital pathology: An overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives. *Journal of Pathology Informatics*, 9(40), 2018.

[11] Tiffany L. Sellaro, Robert Filkins, Chelsea Hoffman, Jeffrey L. Fine, Jon Ho, Anil V. Parwani, Liron Pantanowitz, and Michael Montalto. Relationship between magnification and resolution in digital pathology systems. *Journal of Pathology Informatics*, 4(21), 2013.

[12] Mark D. Zarella, Douglas Bowman, Famke Aeffner, Navid Farahani, Albert Xthona, Syeda Fatima Absar, Anil Parwani, Marilyn Bui, and Douglas J. Hartman. A practical guide to whole slide imaging: A white paper from the digital pathology association. *Archives of Pathology & Laboratory Medicine*, 143(2):222–234, 2019.

[13] Hamid R. Tizhoosh and Morteza Babaie. Representing medical images with encoded local projections. *IEEE Trans. Biomed. Eng.*, 65(10):2267–2277, 2018.

[14] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[15] Xiang Zhou, Sonja Zillner, Manuel Möller, Michael Sintek, Yiqiang Zhan, Arun Krishnan, and Alok Gupta. Semantics and CBIR: A medical imaging perspective. In

Proc. International Conference on Content-based Image and Video Retrieval (CIVR 2008), pages 571–580, Dublin, Ireland, 2008.

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[18] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Proc. Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, Prague, Czech Republic, 2004.

[19] Meghana Dinesh Kumar, Morteza Babaie, Shujn Zhu, Shivam Kalra, and Hamid R. Tizhoosh. A comparative study of CNN, BoVW and LBP for classification of histopathological images. In *Proc. 2017 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2017)*, pages 1–7, Honolulu, USA, 2017.

[20] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[21] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–0893, San Diego, USA, 2005.

[22] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[23] Xueming Qian, Xian-Sheng Hua, Ping Chen, and Liangjun Ke. PLBP: An effective local binary patterns texture descriptor with pyramid representation. *Pattern Recognition*, 44(10):2502–2515, 2011.

[24] Ofir Pele and Michael Werman. The quadratic-chi histogram distance family. In *ECCV 2010, Part II, LNCS 6312*, pages 749–762. Springer-Verlag, 2010.

[25] Li-Yu Hu, Min-Wei Huang, Shih-Wen Ke, and Chih-Fong Tsais. The distance function effect on k-nearest neighbor classification for medical datasets. *SpringerPlus*, 5(1302), 2016.

[26] Franklin Mendivil. Computing the Monge-Kantorovich distance. *Computational and Applied Mathematics*, 36(3):1389–1402, 2017.

[27] Ursula Molter, Jonathan Brandt, and Carlos Cabrelli. An algorithm for the computation of the Hutchinson distance. *Information Processing Letters*, 40(2):113–117, 1991.

[28] Henning Müller, Nicolas Michoux, David Bandon, and Antoine Geissbuhler. A review of content-based image retrieval systems in medical applications–clinical benefits and future directions. *International Journal of Medical Informatics*, 73(1):1–23, 2004.

[29] Harshita Sharma, Alexander Alekseychuk, Peter Leskovsky, Olaf Hellwich, R.S. Anand, Norman Zerbe, and Peter Hufnagl. Determining similarity in histological images using graph-theoretic description and matching methods for content-based image retrieval in medical diagnostics. *Diagnostic Pathology*, 7(134):1–20, 2012.

[30] Xin Qi, Daihou Wang, Ivan Rodero, Javier Diaz-Montes, Rebekah H. Gensure, Fuyong Xing, Hua Zhong, Lauri Goodell, Manish Parashar, David J. Foran, and Lin Yang. Content-based histopathology image retrieval using cometcloud. *BMC Bioinformatics*, 15(287):1–17, 2014.

[31] Shujin Zhu, Yuehua Li, Shivam Kalra, and Hamid R. Tizhoosh. Multiple disjoint dictionaries for representation of histopathology images. *Journal of Visual Communication and Image Representation*, 55:243–252, 2018.

[32] Jorge A. Vanegas, John Arevalo, and Fabio A. González. Unsupervised feature learning for content-based histopathology image retrieval. In *Proc. 12th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, Klagenfurt, Austria, June 2014.

[33] Akshay Sridhar, Scott Doyle, and Anant Madabhushi. Content-based image retrieval of digitized histopathology in boosted spectrally embedded spaces. *Journal of Pathology Informatics*, 6(41), 2015.

[34] Rachel Sparks and Anant Madabhushi. Out-of-sample extrapolation utilizing semi-supervised manifold learning (OSE-SSL): Content based image retrieval for histopathology images. *Scientific Reports*, 6:1–15, 2016.

[35] Meghana Dinesh Kumar, Morteza Babaie, and Hamid R. Tizhoosh. Deep barcodes for fast retrieval of histopathology scans. In *Proceedings of the IEEE World Congress on Computational Intelligence (IEEE WCCI)*, Rio de Janeiro, Brazil, 2018.

[36] Brady Kieffer, Morteza Babaie, Shivam Kalra, and Hamid R. Tizhoosh. Convolutional nueral networks for histopathology image classification: Training vs. using pre-trained networks. In *Proc. International Conference on Image Processing Theory, tools and Applications (IPTA)*, Montreal, Canada, 2017.

[37] Narayan Hegde, Jason D. Hipp, Yun Liu, Michael E. Buck, Emily Reif, Daniel Smilkov, Michael Terry, Carrie J. Cai, Mahul B. Amin, Craig H. Mermel, Philip Q. Nelson, Lily H. Peng, Gregory S. Corrado, and Martin C. Stumpe. Similar image search for histopathology: SMILY. *npj Digital Medicine*, 2(56), 2019.

[38] Hamid R. Tizhoosh, Shujin Zhu, Hanson Lo, Varun Chaudhari, and Tahmid Mehdi. Minmax radon barcodes for medical image retrieval. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Fatih Porikli, Sandra Skaff, Alireza Entezari, Jianyuan Min, Daisuke Iwai, Amela Sadagic, Carlos Scheidegger, and Tobias Isenberg, editors, *Advances in Visual Computing*, pages 617–627, Las Vegas, Nevada, 2016. Springer International Publishing.

[39] Niccolò Marini, Manfredo Atzori, Sebastian Otálora, Stephane Marchand-Maillet, and Henning Müller. H&E-adversarial network: A convolutional neural network to learn stain-invariant features through hematoxylin & eosin regression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 601–610. IEEE Computer Society Conference Publishing Services, 2021.

[40] Arnout C. Ruifrok and Dennis A. Johnston. Quantification of histochemical staining by color deconvolution. *Analytical and Quantitative Cytology and Histology*, 23(4):291–299, 2001.

[41] Michael T. McCann, Joshita Majumdar, Cheng Peng, Carlos A. Castro, and Jelena Kovacevic. Algorithm and benchmark dataset for stain separation in histology images. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 3953–3957, Paris, France, 2014.

[42] Marc Macenko, Marc Niethammer, J. S. Marron, David Borland, John T. Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E. Thomas. A method for normalizing histology slides for quantitative analysis. In *Proc. of IEEE International Symposium on Biomedical Imaging (ISBI): From Nano to Macro*, pages 1107–1110, Chicago, USA, 2009.

[43] Fabio A. Spanhol, Luiz S. Oliveira, Caroline Petitjean, and Laurent Heutte. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, 63(7):1455–1462, 2016.

[44] Brian C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77:157–173, 2008.

[45] Zhongheng Zhang. Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, 4(11), 2016.

[46] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 1977.

[47] Hongping Hu, Shichang Qiao, Yan Hao, Yanping Bai, Rong Cheng, Wendong Zhang, and Guojun Zhang. Breast cancer histopathological images recognition based on two-stage nuclei segmentation strategy. *PLoS ONE*, 17(4), 2022.

[48] Benoît B. Mandelbrot. *Les Objets Fractals: Forme, Hasard et Dimension.* Paris: Flammarion, 1975.

[49] Benoît B. Mandelbrot. *The Fractal Geometry of Nature.* W.H. Freeman and Company, 1983.

[50] Jens Feder. *Fractals.* Springer, 1988.

[51] Kenneth Falconer. *Fractal Geometry: Mathematical Foundations and Applications.* John Wiley & Sons Ltd., 2014.

[52] Michael F. Barnsley and Stephen Demko. Iterated function systems and the global construction of fractals. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 399(1817):243–275, 1985.

[53] John E. Hutchinson. Fractals and self similarity. *Indiana University Mathematics Journal*, 30(5):713–747, 1981.

[54] Heinz-Otto Peitgen, Hartmut Jürgens, Dietmar Saupe, Evan Maletsky, Terry Perciante, and Lee Yunker. *Fractals for the Classroom: Strategic Activities Volume One.* Springer-Verlag, 1991.

[55] Michael F. Barnsley. *Fractals Everywhere, Second Edition.* Academic Press Professional, 1993.

[56] Roman Duda. The origins of the concept of dimension. *Colloquium Mathematicum*, 42(1):95–110, 1979.

[57] Saber Elaydi. *Discrete Chaos, Second Edition: With Applications in Science and Engineering*. Taylor & Francis Group, LLC, 2007.

[58] Gerald A. Edgar. *Measure, Topology and Fractal Geometry*. Springer-Verlag, 1990.

[59] H. George E. Hentschel and Itamar Procaccia. The infinite number of generalized dimensions of fractals and strange attractors. *Physica D: Nonlinear Phenomena*, 8(3):435–444, 1983.

[60] Ashvin Chhabra and Roderick V. Jensen. Direct determination of the $f(\alpha)$ singularity spectrum. *Physical Review Letters*, 62(12):1327–1330, 1989.

[61] Tamás Tél, Agnes Fülöp, and Tamás Vicsek. Determination of fractal dimensions for geometrical multifractals. *Physica A*, 159(2):155–166, 1989.

[62] Tamás Vicsek. Mass multifractals. *Physica A*, 168:490–497, 1990.

[63] Tamás Vicsek. Multifractal geometry of diffusion-limited aggregates. *Europhysics Letters*, 12(3):217–222, 1990.

[64] Martin Bouda, Joshua S. Caplan, and James E. Saiers. Box-counting dimension revisited: Presenting an efficient method of minimizing quantization error and an assessment of the self-similarity of structural root systems. *Frontiers in Plant Science*, 7:1–15, 2016.

[65] J. M. Halley, S. Hartley, A. S. Kallimanis, W. E. Kunin, J. J. Lennon, and S. P. Sgardelis. Uses and abuses of fractal methodology in ecology. *Ecology Letters*, 7:254–271, 2004.

[66] Rolf Riedi. An improved multifractal formalism and self-similar measures. *Journal of Mathematical Analysis and Applications*, 189(2):462–490, 1995.

[67] Giorgio Mantica. The global statistics of return times: Return time dimensions versus generalized measure dimensions. *Journal of Statistical Physics*, 138:701–727, 2010.

[68] Joseph Panico and Peter Sterling. Retinal neurons and vessels are not fractal but space-filling. *The Journal of Comparative Neurology*, 361:479–490, 1995.

[69] Yuval Gazit, James W. Baish, Nina Safabakhsh, Michael Leunig, Laurence T. Baxter, and Rakesh K. Jain. Fractal characteristics of tumor vascular architecture during tumor growth and regression. *Microcirculation*, 4(4):395–402, 1997.

[70] Ethel Nilsson. Multifractal-based image analysis with applications in medical imaging. Master's thesis, Umeå University, Umeå, Sweden, 2007.

[71] Tomislav Stojić, Irini Reljin, and Branimir Reljin. Adaptation of multifractal analysis to segmentation of microcalcifications in digital mammograms. *Physica A*, 367:494–508, 2006.

[72] J.F. Muzy, E. Bacry, and A. Arneodo. The multifractal formalism revisited with wavelets. *International Journal of Bifurcation and Chaos*, 4(2):245–302, 1994.

[73] Pierre Kestener, Jean Marc Lina, Philippe Saint-Jean, and Alain Arneodo. Wavelet-based multifractal formalism to assist in diagnosis in digitized mammograms. *Image Analysis and Stereology*, 20:169–174, 2001.

[74] 3Blue1Brown. Fractals are typically not self similar. https://youtu.be/gB9n2gHsHN4?t=940, 2017. Accessed on: 2022-06-01.

[75] Russell Reeve. A warning about standard errors when estimating the fractal dimension. *Computers & Geosciences*, 19:89–91, 1992.

[76] Sylvie Lorthois and Francis Cassot. Fractal analysis of vascular networks: Insights from morphogenesis. *Journal of Theoretical Biology*, 262:614–633, 2010.

[77] Paola La Barbera and Renzo Rosso. On the fractal dimension of stream networks. *Water Resources Research*, 25:735–741, 1989.

[78] Samuele G. De Bartolo, Roberto Gaudio, and Salvatore Gabriele. Multifractal analysis of river networks: Sandbox approach. *Water Resources Research*, 40:1–10, 2004.

[79] James W. Baish and Rakesh K. Jain. Fractals and cancer. *Cancer Research*, 60:3683–3688, 2000.

[80] James D. Murray. Use and abuse of fractal theory in neuroscience. *The Journal of Comparative Neurology*, 361:369–371, 1995.

[81] Daniel Hamburger, Ofer Biham, and David Avnir. Apparent fractality emerging from models of random distributions. *Physical Review E*, 53:3342–3358, 1996.

[82] Raffi Karshafian, Peter N. Burns, and Mark R. Henkelman. Transit time kinetics in ordered and disordered vascular trees. *Physics in Medicine and Biology*, 48:3225–3237, 2003.

[83] Jian-Long Kou, Hang-Jun Lu, Feng-Min Wu, and You-Sheng Xu. Sprout branching of tumour capillary network growth: Fractal dimension and multifractal structure. *Chinese Physics Letters*, 25:1746–1749, 2008.

[84] Daniele Mancardi, Gianfranco Varetto, Enrico Bucci, Fabrizio Maniero, and Caterina Guiot. Fractal parameters and vascular networks: Facts & artifacts. *Theoretical Biology and Medical Modelling*, 5:1–8, 2008.

[85] Yuval Gazit, David A. Berk, Michael Leunig, Laurence T. Baxter, and Rakesh K. Jain. Scale-invariant behavior and vascular network formation in normal and tumor tissue. *Physical Review Letters*, 75(12):2428–2431, 1995.

[86] Yuval Gazit. *Fractal Vasculature and Vascular Network Growth Modeling in Normal and Tumor Tissue*. PhD thesis, Massachusetts Institute of Technology, 1996.

[87] Laurent Risser, Franck Plouraboue, Alexandre Steyer, Peter Cloetens, Géraldine Le Duc, and Caroline Fonta. From homogeneous to fractal normal and tumorous microvascular networks in the brain. *Cerebral Blood Flow & Metabolism*, 27:293–303, 2007.

[88] Loretta Ichim and Radu Dobrescu. Characterization of tumor angiogenesis using fractal measures. In Ioan Dumitrache, Adina Magda Florea, and Florin Pop, editors, *Proc. 19th International Conference on Control Systems and Computer Science, CSCS 2013*, pages 345–349, Bucharest, Romania, 2013.

[89] Naomi Tsafnat, Guy Tsafnat, and Tim D. Lambert. A three-dimensional fractal model of tumour vasculature. In *Proc. 26th Annual International Conference of the IEEE EMBS*, pages 683–686, San Fransisco, USA, 09 2004.

[90] Fereydoon Family, Barry R. Masters, and Daniel E. Platt. Fractal pattern formation in human retinal vessels. *Physica D*, 38:98–103, 1989.

[91] Martin A. Mainster. The fractal properties of retinal vessels: Embryological and clinical implications. *Eye*, 4:235–241, 1990.

[92] Daniel J. Gould, Tegy J. Vadakkan, Ross A. Poché, and Mary E. Dickinson. Multifractal and lacunarity analysis of microvascular morphology and remodeling. *Microcirculation*, 18:136–151, 2011.

[93] Ştefan Ţălu, Sebastian Stach, Dan Mihai Călugăru, Carmen Alina Lupaşcu, and Simona Delia Nicoară. Analysis of normal human retinal vascular network architecture

using multifractal geometry. *International Journal Of Ophthalmology*, 10:434–438, 2017.

[94] Tatijana Stošić and Borko D. Stošić. Multifractal analysis of human retinal vessels. *IEEE Transactions on Medical Imaging*, 25:1101–1107, 2006.

[95] Edmund Arthur, Gabor Mark Somfai, Maja Kostic, Susel Oropesa, Carlos Mendoza Santiesteban, and Delia Cabrera DeBuc. Distinguishing cognitive impairment by using singularity spectrum and lacunarity analysis of the retinal vascular network. *Neurophotonics*, 6(4):1–8, 2019.

[96] Ruofan Wang, Peilun Li, John K.-J. Li, Wolfgang M. Kuebler, and Axel R. Pries. Investigation into the diversity in the fractal dimensions of arterioles and venules in a microvascular network - a quantitative analysis. *Microvascular Research*, 125:1–10, 2019.

[97] Fadilah Ab Hamid, Mohd Zulfaezal Che Azemin, Adzura Salam, Amilia Aminuddin, Norsyazwani Mohd Daud, and Ilyanoon Zaharai. Retinal vasculature fractal dimension measures vessel density. *Current Eye Research*, 41:823–831, 2016.

[98] Audrey Karperien. Fraclac for ImageJ. https://imagej.nih.gov/ij/plugins/fraclac/FLHelp/Introduction.htm, 1999–2013. Accessed on: 2020-02-01.

[99] STructured Analysis of the REtina Project. http://cecas.clemson.edu/~ahoover/stare/. Accessed on: 2022-04-21.

[100] Adam Hoover, Valentina Kouznetsova, and Michael Goldbaum. Locating blood vessels in retinal images by piece-wise threshold probing of a matched filter response. *IEEE Transactions on Medical Imaging*, 19(3):203–210, 2000.

[101] Luna B. Leopold. Trees and streams: The efficiency of branching patterns. *Journal of Theoretical Biology*, 31:339–354, 1971.

[102] Juan M. García-Ruiz and Fermín Otálora. Fractal trees and horton's laws. *Mathematical Geology*, 24(1):61–71, 1992.

[103] A Bejan. *Shape and Structure, from Engineering to Nature*. Cambridge University Press, 2000.

[104] Jonathan M. Fraser. Inhomogeneous self-similar sets and box dimensions. *Studia Mathematica*, 213:133–156, 2012.

APPENDICES

Appendix A

The Box-Counting Dimension of a Simple Fractal Tree and its Canopy

In this section we derive an expression for the box-counting dimension of a simple fractal tree, T. For ease of notation, we shall use D to refer to the box-counting dimension, $dim_B(T)$, of the tree throughout this Appendix.

Consider the tree T, as shown in Figure A.1, with $N = 2$, $\theta_1 = \theta_2 = \theta$, and $r_1 = r_2 = r$ and let the length of the trunk be given by b. First we compute the total length L of the set T to see if/where a fractal dimension is necessary. From Figure A.1 we see that the length can be expressed in terms of the trunk-length b as a geometric series,

$$\begin{aligned} L &= b + 2rb + 4r^2b + 8r^3b + \cdots \\ &= b[1 + 2r + 4r^2 + 8r^3 + \cdots]\,. \end{aligned} \tag{A.1}$$

The above series converges for $0 \leq r < \frac{1}{2}$, in which case,

$$L = \frac{b}{1 - 2r}\,. \tag{A.2}$$

The length L is finite for $1 \leq r < \frac{1}{2}$ and infinite for $r \geq \frac{1}{2}$, which tells us that the dimension D of T is 1 for $r < \frac{1}{2}$, i.e.,

$$\boxed{D = 1\,, \quad 0 \leq r < \frac{1}{2}\,.} \tag{A.3}$$

Since the length of T is infinite for $r \geq \frac{1}{2}$, we expect that $D > 1$ for $\frac{1}{2} < r < 1$ and that the case $r = \frac{1}{2}$ will have to be considered separately. In the following discussion we assume

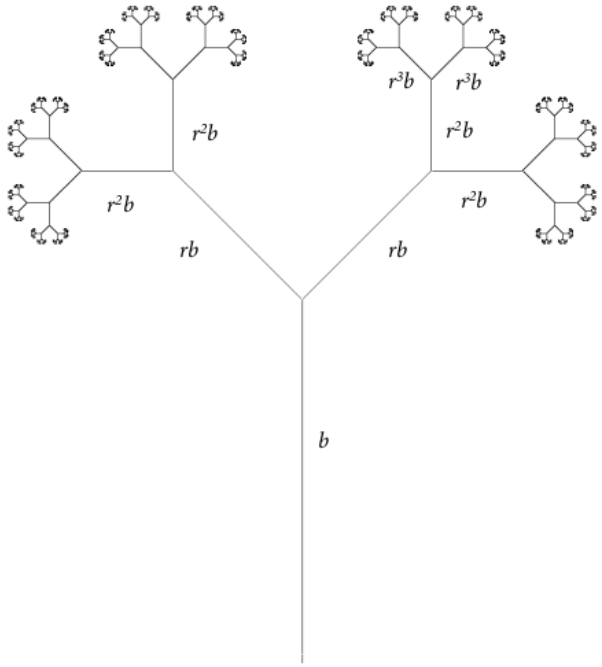

Figure A.1: A fractal tree, T, with $N = 2$, a single scale factor r, and trunk length b.

values of r sufficiently low such such that the two primary subtrees of T do not overlap.

The set T can be considered as a union of its trunk and its two major subtrees, i.e.,

$$T = T_1 \cup T_2 \,. \tag{A.4}$$

We'll let $N_1(\epsilon)$ and $N_2(\epsilon)$ be the number of square tiles of side length ϵ needed to cover T_1 and T_2, respectively. The number of ϵ-tiles needed to cover the trunk is given exactly by

$$N_1(\epsilon) = \frac{b}{\epsilon} \,. \tag{A.5}$$

so that

$$N(\epsilon) = N_1(\epsilon) + N_2(\epsilon) = \frac{b}{\epsilon} + N_2(\epsilon) \,. \tag{A.6}$$

is the number of ϵ-tiles needed to cover T. Now, we consider covering T_2 with tiles of side length $r' = r\epsilon$. Since each subtree is an r-scaled copy of T, we have the following:

$$N_2(r\epsilon) = 2N(\epsilon) \,, \tag{A.7}$$

and the total number of tiles of side length $r' = r\epsilon$ needed to cover the entire set T is given by

$$\boxed{N(r\epsilon) = \frac{b}{r\epsilon} + 2N(\epsilon) \,.} \tag{A.8}$$

We'll now employ the basic scaling relation,

$$N(r\epsilon) \approx N(\epsilon) r^{-D} \,, \quad \text{as} \quad \epsilon \to 0^+ \,. \tag{A.9}$$

Substitution of (A.9) into (A.8) yields

$$N(\epsilon) r^{-D} \approx 2N(\epsilon) + \frac{b}{r\epsilon} \quad \text{as} \quad \epsilon \to 0^+ \,. \tag{A.10}$$

Now divide both sides by $N(\epsilon)$,

$$r^{-D} \approx 2 + \frac{b}{r\epsilon N(\epsilon)} \quad \text{as} \quad \epsilon \to 0^+ \,. \tag{A.11}$$

If the set T has box-counting dimension D, then

$$N(\epsilon) \approx B\epsilon^{-D} \quad \text{as} \quad \epsilon \to 0^+ \,, \tag{A.12}$$

where $B > 0$ is a constant. (It is, in fact, this relation that is responsible for the basic scaling relation in Equation (A.9).) Substitution of this relation into (A.11) yields,

$$\boxed{r^{-D} \approx 2 + \frac{b}{Br}\epsilon^{D-1} \,, \quad \text{as} \quad \epsilon \to 0^+ \,.} \tag{A.13}$$

- **Case No. 1:** $D > 1$. Then $\epsilon^{D-1} \to 0$ as $\epsilon \to 0^+$ so that

$$r^{-D} = 2 \quad \Longrightarrow \quad \boxed{D = \frac{\log 2}{\log\left(\frac{1}{r}\right)} \,, \quad \frac{1}{2} < r < r^* \,,} \tag{A.14}$$

  where $r^* < 1$ is the value of r at which the two subtrees of T intersect. (r^* depends upon the value of θ defining the generator of T. A discussion of this relationship is beyond the scope of this paper.) Note that in the limit $r \to \frac{1}{2}$, $D \to 1$. We expect that Equation (A.14) holds in the case $r = \frac{1}{2}$.

- **Case No. 2:** $D = 1$. For all $\epsilon \to 0^+$,

$$\frac{1}{r} = 2 + \frac{b}{Br} \,, \tag{A.15}$$

  which can be rearranged to give the following result,

$$B = \frac{b}{(1-2r)} \,, \quad 0 < r < \frac{1}{2} \,. \tag{A.16}$$

  But from Equation (A.2), we have

$$B = L \,. \tag{A.17}$$

  Substitution of this result into Equation (A.12) yields

$$N(\epsilon) = \frac{L}{\epsilon} \,, \tag{A.18}$$

  which makes sense: When $D = 1$, the set T has finite length L. The number of ϵ-tiles needed to cover T is given by Equation (A.18).

Summary of results: The fractal dimension D of the set T is

$$\boxed{D(r) = \begin{cases} 1, & 0 \le r \le \frac{1}{2}\,, \\ \frac{\log 2}{\log 1/r}\,, & \frac{1}{2} \le r < r^*\,. \end{cases}} \tag{A.19}$$

Note that the above result can easily be extended to the more general case of $N \ge 2$ branches. The $\log 2$ in Equation (A.19) is simply replaced by $\log N$.

The **canopy** C of the fractal tree T is the set of all limit points of the sequences of endpoints of the branches generated by the infinite application of the branching generator – essentially the set of all "tips" of the tree T. The canopy of the fractal tree T of Figure A.1 is shown in Figure A.2.

Figure A.2: The canopy of T.

Each of the two "halves" of the canopy C shown in Figure A.2 is a contracted and translated copy of the set C. The contraction factor is r. As such, C is a self-similar fractal. The usual scaling argument for such self-similar fractals yields the following result for the dimension D_C of the canopy C:

$$D_C(r) = \frac{\log 2}{\log 1/r}\,, \quad 0 < r < r^*. \tag{A.20}$$

This may be combined with Equation (A.19) to yield the following interesting result,

$$D(r) = \max\{D_C(r), 1\}\,, \quad 0 < r < r^*\,. \tag{A.21}$$

In other words, if the fractal tree T is truly fractal à la Mandelbrot, i.e., $D > 1$, then $D = D_C$, the fractal dimension of its canopy. Equation (A.21) is a particular case of a more general result for box-counting dimensions of inhomogeneous self-similar sets [104]. Finally, we mention that in the case of unskeletonized trees, i.e. where the trunk and branches are "thick" and therefore two-dimensional, the dimension of T is

$$D(r) = \max\{D_C(r), 2\}\,, \quad 0 < r < r^*\,. \tag{A.22}$$