Wrocław University of Science and Technology
Department of Computer Engineering
W–4 / K–9

Wrocław University
of Science and Technology

Report of the PRE series No W04 / 2019 / P–053

Diagnostic methods based on the analysis of interference images

(doctoral dissertation)

Metody diagnostyczne oparte na analizie obrazów interferencyjnych

(rozprawa doktorska)

Piotr Szyperski

Supervisor: prof. dr hab. inż. Ewa Skubalska-Rafajłowicz
Second supervisor: dr hab. inż. D. Robert Iskander, prof. PWr

Keywords: fractal dimension,
box-counting,
interferometry,
diagnostics

Wrocław
Poland

2019
September

To those without whom this work would not have been completed:
to my beloved parents, Irena and Dariusz,
to my grandmothers, Marianna and Ewa,
and to my love, Janka.
Thank you.

— P.

ACKNOWLEDGMENTS

The author expresses gratitude to his Ph.D. advisors, prof. dr hab. inż. Ewa Skubalska-Rafajłowicz and dr hab. inż. D. Robert Iskander, prof. PWr, for their invaluable help in bringing this thesis into being.

The author remains thankful for the guidance, encouragement and advice they have provided him as well as for the and trust and scientific and personal support they have granted him in the course of the doctoral research with all its challenges and demands. Further, the author appreciates the value of abundant and fruitful discussions, multiple prolific and inspiring suggestions together with the assistance of adroit proofreading eye during the process of improving numerous revisions of the manuscript of this dissertation.

You're a refraction of the one light.
You're a waveform of light.
You're a fractal, a pattern that continuously changes.

— Frederick P. Lenz

STRESZCZENIE

Istnieje wiele zastosowań biomedycznych, w których analiza obrazów interferencyjnych może doprowadzić do lepszych metod diagnostycznych. Obrazy tego typu mogą być traktowane jako tekstury i stąd estymacja wymiaru fraktalnego obrazu w skali szarości jest szczególnie interesująca. Istnieje wiele podejść do obliczania wymiaru fraktalnego obrazu, w tym algorytm binarnego zliczania pudełek i jego różnicowe rozszerzenia jak również liczne inne deskryptory obrazu wykorzystujące fraktalność. Niemniej jednak obecnie istniejący zestaw estymatorów wymiaru fraktalnego nie wyczerpuje możliwości opracowania innych typów estymatorów, które w konkretnych zastosowaniach przewyższałyby wcześniej stosowane rozwiązania. Estymator wymiaru fraktalnego może być lepszy niż inne pod względem właściwości statystycznych takich jak wariancja lub dokładność określenia wartości oczekiwanej, wydajność diagnostyki (o ile znajduje tego typu zastosowanie) lub wydajność obliczeniowa. W zależności od oczekiwanego zastosowania, jedną lub więcej spośród wspomnianych właściwości można przyjąć jako kryterium oceny opracowanych metod estymacji oraz kryterium porównawcze między nimi. Niniejsza praca doktorska ma dwa główne cele. Pierwszym z nich jest opracowanie estymatorów wymiaru fraktalnego dla dwuwymiarowych obrazów w skali szarości, które spełniają następujące warunki: odpowiednią wydajność obliczeń numerycznych, wysoki zakres dynamiki oraz zgodność z wymiarem euklidesowym (topologicznym) dla zbiorów niebędących fraktalami (w sensie samopodobieństwa). Drugim jest zastosowanie nowo zaprojektowanych estymatorów wymiaru fraktalnego do zagadnienia interferometrycznego obrazowania ludzkiego filmu łzowego. Celami cząstkowymi pracy są: badanie właściwości estymatorów (opracowanie względem istniejących rozwiązań) oraz rozważenie nowych podejść do estymacji wymiaru fraktalnego, to jest probabilistycznych w stosunku do nieprobabilistycznych. Rozważane są dwa nowe podejścia do estymacji wymiaru fraktalnego obrazów w skali szarości opartej na podejściu pudełkowym, aby przezwyciężyć pewne ograniczenia standardowej metody pudełkowej. Obejmują one estymatory wymiaru fraktalnego na podstawie ważonej metody pudełkowej w skali szarości oraz probabilistycznej metody pudełkowej w skali szarości opartej na prawdopodobieństwie zdefiniowanym w przestrzeni

obrazu. Są one zestawione z algorytmami estymacji wymiaru fraktalnego wykorzysującymi standardową binarną metodę pudełkową oraz probabilistyczną metodę pudełkową w skali szarości opartą na prawdopodobieństwie zdefiniowanym w przestrzeni intensywności. Zarówno obrazy syntetyczne jak i interferogramy pozyskane empirycznie zostały użyte do oceny proponowanych estymatorów wymiaru fraktalnego, ze szczególnym naciskiem położonym na ich zdolności diagnostyczne. W pierwszej kolejności obrazy syntetyczne obejmujące pełen zakres wartości wymiaru fraktalnego zostały wykorzystane do walidacji proponowanych estymatorów wymiaru fraktalnego. Następnie interferometryczne obrazy w skali szarości przedstawiające film łzowy osób zdrowych oraz pacjentów dotkniętych zespołem suchego oka zastosowano do oceny mocy dyskryminacyjnej diagnostyki suchego oka opartej na wymiarze fraktalnym. Wykonane badania zakończone zostały uzyskaniem dwóch głównych wyników. Pierwszym jest rozwój teorii dotyczącej dwóch nowych estymatorów wymiaru fraktalnego, które okazały się lepsze od wcześniej rozważanych. Kolejny to opracowanie algorytmów służących diagnostyce suchego oka na podstawie analizy wymiaru fraktalnego obrazów interferometrii metodą poprzecznego przesunięcia czoła fali pozyskanych in-vivo. Klasyfikacja obrazów z interferometru poprzecznego przesunięcia czoła fali oparta na wymiarze fraktalnym porównana została z fourierowskim podejściem spektralnym, gdzie okazała się równie efektywna pod względem charakterystyk operacyjnych odbiorników jednakże znacznie wydajniejsza obliczeniowo. Podsumowując – ogólnym celem pracy było zbadanie opartych na fraktalności deskryptorów parametrów obrazu w odniesieniu do jakościowej oceny jego struktury oraz – w przypadku sekwencji wideo – opisanie przebiegu ich dynamicznych zmian. Wyniki obejmują nowe estymatory jakości danych interferometrycznych. Niektóre z metod analizy obrazów opracowano poprzez samodzielny projekt i implementację, np. algorytmu ważonej metody pudełkowej w skali szarości, rekurencyjne podejście do decymacji próbek lub równoległa implementacja metody Saupego przeskaluj-i-dodaj do generowania syntetycznych fraktali przyspieszającej obliczenia dziesiątki tysięcy razy. U podstaw potrzeby motywującej prowadzenie badań legły pragmatyczne przesłanki, jednak – pomijając wartość czysto użytkową – badania te również wniosły przyczynek do rozwoju metodologii estymacji właściwości i cech fraktalnych. Praktyczne znaczenie badań deskryptorów obrazu opartych na fraktalności uzasadnia badania teoretyczne oraz doświadczalne jako cechujące się techniczną przydatnością przy zastosowaniach w dziedzinach diagnostyki biomedycznej oraz przemysłowej. Przy opisie całościowego znaczenia przeprowadzonych badań należy zauważyć, iż rozwój

oraz poprawa działania zaproponowanych metod estymacji wymiaru fraktalnego warte są wkładu czasu oraz nakładu sił. W szczególności zaś wyniki przeprowadzonych badań empirycznych wysunąć można jako wydatny przykład potwierdzający użyteczność wspomnianego rozszerzonego studium. Stąd też w pewnych aplikacjach nowo zaproponowane deskryptory oparte na fraktalności zdolne działać w skali szarości okazały się lepsze od innych algorytmów wykorzystujących wymiar fraktalny oraz porównywalne z technikami opartymi na metodach spektralnych – jeśli nie lepsze od nich. Jednocześnie można zaimplementować te metody w sposób pozwalający na zapewnienie podobnej lub lepszej wydajności obliczeniowej. Oba te aspekty stanowią silną podstawę do rozważenia przeprowadzenia w przyszłości dodatkowych badań rozwijających wspomniane zagadnienia.

ABSTRACT

There are many biomedical applications in which the analysis of interference images can lead to better diagnostic methods. Such images can be treated as textures and for that fractal dimension estimation over grayscale image is of particular interest. There are different approaches to calculating fractal dimension of an image, including binary box-counting algorithm and its differential extensions as well as the multitude of other fractal-based image descriptors. However, the currently existing set of fractal dimension estimators does not exhaust the possibility of developing other types of estimators that would outperform their predecessors in particular applications. An estimator of fractal dimension can be superior among others in terms of statistical properties such as variance or bias, diagnostic performance (if used in such an application) or computational efficiency. Depending on the desired application, one or more of the mentioned characteristics may be chosen as a criterion for assessment of the developed estimation methods as well as comparison between them. There are two main goals of this doctoral work. The first one is the construction of fractal dimension estimators for two-dimensional grayscale images that satisfy the following conditions: sufficient numerical efficiency, high dynamic range, and consistency with Euclidean (topological) dimension for non-fractal (in a self-similarity sense) sets. The second one is the applicability of the newly designed fractal dimension estimators to the problem of interferometric imaging of tear film in human subjects. The specific objectives of the work are: investigating properties of fractal dimension estimators (research in terms of existing solutions) and considering new approaches to fractal dimension estimation, i.e., probabilistic vs non-probabilistic. Two new approaches for box-counting-based fractal dimension estimation for gray-scale images are considered to overcome some limitations of standard box-counting method. They include weighted grayscale box-counting and probabilistic grayscale box-counting based on image probability space fractal dimension estimators. They are contrasted against the standard binary box-counting as well as probabilistic grayscale box-counting based on intensity probability space fractal dimension estimation algorithms. Both synthetic images as well as real interferometric ones have been used to evaluate the proposed estimators of fractal dimen-

sion, particularly in the view of their diagnostic ability. First, synthetic images that cover a range of fractal dimension values have been used to validate the proposed estimators of fractal dimension. Second, gray-level interferometric images of tear film from normal and dry eye syndrome-affected subjects have been used to assess the discriminative power of fractal-dimension-based diagnosis of dry eye. There are two main outcomes of this study. The first one is the theoretical development of two new fractal dimension estimators that have been found to be superior to the previously considered ones. The second one is the development of algorithms for diagnosing dry eye based on fractal dimension analysis of lateral shearing interferometry images acquired in-vivo. Fractal-dimension--based classification of lateral shearing interferometry images has been compared to Fourier-based spectral approach and has shown equivalent performance in terms of receiver operating characteristics, but much higher computational efficiency. Concluding, the overall goal of the work has been to investigate fractal-based descriptors of image parameters with respect to qualitative assessment of its structure and – in case of video sequences – to describe the course of their dynamic changes. The result consists of new estimators of interferogram data quality. Some of the image analysis methods have been achieved with own design and implementation, e.g., weighted box-counting algorithm for grayscale images, the recursive downsampling approach or the parallelized implementation of Saupe's rescale-and-add method for generating synthetic fractals that has accelerated the computations tens of thousands times. The need behind the conducted research has arisen under pragmatic premises, however – apart from utilitarian circumstances – it also forms a cause for development of estimation of fractal features. The practical significance of studies on fractal-based image descriptors justifies the theoretical and experimental research as exhibiting technical applicability in fields of biomedical and industrial diagnostics. Summing up the significance of the study, it is to be noted that there is a worth in investing the time and effort in the development and enhancement of the proposed fractal dimension estimation methods. In particular, empirical results of this study may be excerpted as a notable example provisioning the utility of such extended research. Hence, in certain applications the newly proposed fractal-based descriptors operating in grayscale have been found to be superior among other fractal-based algorithms and comparable – if not better – with techniques based on spectral methods. At the same time, they may be implemented to provide similar or better computational efficiency. These two aspects are strong advocates behind additional research studies on this matter in the future.

TABLE OF CONTENTS

Part I

INTRODUCTION

Chapter 1 is based mostly on the literature review. However, new conceptualization of the notation convention in chapter 1 contains own contributions, as it prepares the ground for the proposed algorithms to be introduced. It is of importance also because it allows to distinguish between two notations developed to serve different (theoretical vs applicative) purposes.

1 FUNDAMENTALS OF FRACTAL DIMENSION ESTIMATION

1.1 INTRODUCTORY LITERATURE REVIEW WITH THEORETICAL BACKGROUND

1.1.1 General concept of fractal

A fractal description of natural objects has been introduced by *Benoît B. Mandelbrot* who used self-similarity to explore morphological properties of common irregular or fragmented shape patterns that tend to be very hard to parametrize in terms of Euclidean geometry found in, e.g., clouds, mountains or tree bark [1]. An extensive description of various aspects of fractal geometry and a wide overview of available techniques has been presented by *Kenneth John Falconer* in numerous publications including several *important books* [2–4]. Additionally, fractal geometry and its relation to measure theory and topology is a subject of the book by *Gerald Arthur Edgar* that aims to bridge the gap between traditional books on topology and analysis and more specialized treatises on fractal geometry [5]. Furthermore, *Michael Fielding Barnsley and Hawley Rising* have thoroughly described fractal-based descriptors of images, theoretical and experimental methods for concept of fractal dimension (FD) determination elaborated below as well as examples of fractal sets and application of invariant fractal-based descriptor in data compression especially for computer graphics usages [6].

There is multitude of synthetic summaries describing mathematical formalisms regarding dimensionality of self-similar† objects. Among them there are some books including the mentioned one by Mandelbrot [1] as well as shorter papers covering the subject with broad overviews like the unified approach proposed by Witold Kinsner [7].

† A *self-similar* object is exactly or approximately similar to a part of itself.

1.1.2 Fractal geometry and dimension

Fractal geometry arises for objects having self-similarity property as defined by Mandelbrot. Thus, there is introduced a dimension measure that equals Euclidean, i.e., topological, dimension of the set for sets being descriptions of ordinary geometric shapes. E.g., FD of (sets that describe) points is zero, for lines it equals to one, for surfaces – two, volumes – three etc. However, for a set of a shape representing the fractal geometry properties this measure exceeds its topological dimension. Such a descriptor of the set is called a theoretical FD and similarly to Euclidean dimension describes this set in a non-unique manner, i.e., one cannot reconstruct a fractal knowing only its FD value.

Definition 1. Fractal dimension:
Fractals (and some ordinary geometrical objects) satisfy scaling rule:

$$N_\epsilon \propto \epsilon^{-D}, \tag{1-1}$$

where D denotes FD of a specific set, N_ϵ – the number (usually its minimal value) of hypermanifolds (sometimes called *rulers* or *boxes*) of topological dimension not smaller than D that are required to form a (usually disjoint) union covering the whole reference set at a given ruler scaling factor ϵ. □

Equation (1-1) from definition 1 can be rewritten as to allow (direct or based on best fit of an linear regression) calculation of D:

$$D \doteq -\log_\epsilon N_\epsilon = \frac{\log N_\epsilon}{\log \epsilon^{-1}}. \tag{1-2}$$

In addition to the number of theoretically described approaches mentioned below, there exists an abundance of ready to use applications of image analysis tools offering FRACTAL DIMENSION estimation (FDE) modules [8, 9].

When FDE is to be utilized as a tool for classification applications, the often-advocated paradigm formulated by Qian Huang, Jacob R. Lorch and Richard C. Dubes in 1994 [10] and sometimes referred to in this context as *the philosophy of Huang* or *the spirit of Huang*, shall be considered. This general approach to the possibilities and applications of the measurements of FD is also pointed out and formulated in chapter 3.

1.2 DESCRIPTION OF DESIGNED BOX-COUNTING METHODS AND ALGORITHMS

1.2.1 Discrete signals and monochromatic binary images

There is a plethora of exact and approximate exemplifications of FD concept that in general do not have to be equivalent for all fractal sets, but should satisfy some form of the scaling rule. Some of them also make use of power law relationship between ruler/box and scale-dependent quantities that are comes around for fractal sets. It is the case mostly for empirical estimators of FD, but when this holds, FD value can be derived directly from linear regression of log–log plot as it is the exponent of properly formulated power law functional relation.

One of the notable examples of FD is the *Hausdorff's–Besicovitch's dimension* (also known in shortened form as *Hausdorff's dimension* or *capacity dimension*) $\dim_{\mathrm{H}}(\cdot)$ that is calculable in rigorous terms only for some analytically described sets and basically for naturally occurring fractals might be only approximated, at least to some degree – in a somewhat heuristic manner [11, 12].

Related to Hausdorff's dimension, but simpler and practically much more applicable for wide class of scenes including natural images notions are, e.g., correlation dimension ν, packing dimension $\dim_{\mathrm{P}}(\cdot)$ and *box-counting (BC) dimension* $\dim_{\mathrm{box}}(\cdot)$ also known as *Minkowski–Bouligand dimension* or *Minkowski dimension*. All these measures give identical values for a typical nature-borne fractal sets, but it is worth noticing that there are objects having all these dimensions inherently different.

BC dimension is employed for FDE particularly common in engineering calculations as well as in natural sciences, i.e., applied to situations where photographs of real objects or patterns are processed to qualitatively characterize their surface or texture features [13, 14]. In its basic variant it is usually implemented to operate on one-dimensional vector or two-dimensional (2-D) binary matrix. One simple definition states that if $N(\epsilon)$ is the number of boxes of side length ϵ required to cover a set A, then BC dimension of that set is defined as:

$$\dim_{\mathrm{box}}(A) \doteq \lim_{\epsilon \to 0^{+}} \frac{\log N(\epsilon)}{\log 1/\epsilon}. \tag{1-3}$$

Often, $N(\epsilon)$ is denoted in shortened form as:

$$N(\epsilon) \doteq N_\epsilon. \tag{1-4}$$

Limit superior or limit inferior may be taken into account if the above is not applicable for some reason. These define the *upper box dimension* (also called *entropy dimension*, Kolmogorov's dimension, Kolmogorov's capacity, limit capacity) and *lower box dimension*, respectively [2, 15]. These terms exhibit a relation to Hausdorff's dimension concept. Contradistinction between the methods pointed out above is not crucial for a fractal set that does not exhibit some very atypical traits and therefore it is at times neglected in descriptions considering their practical applications. Another one similar measure of FD is the aforementioned correlation dimension ν.

1.2.2 Grayscale images

Mandelbrot's conception of a specific length–scale dependency that involves FD coefficient observed for, e.g., coastlines, has been later exploited by Shmuel Peleg et al. [16]. Its applicability has been broadened to allow empirical FD calculation based not only on the length (for a curve) but also on the area (for a surface) by covering it with the ϵ-blanket and determination of its volume. This method tends to saturate slightly for surfaces with high fractal dimensionality, i.e., close to 3, and its computational complexity results in calculation times placing it in the middle of FDE methods set described and compared by Nirupam Sarkar and Bidyut B. Chaudhuri in [17], where they also proposed a new approach to this problem called *differential* BOX-COUNTING (DBC) described below. The ϵ-blanket algorithm needs many iterations for different scales – as many as 15 [17] to 50 [16] – to give accurate result what effectively worsens its performance by a substantial factor.

Alex P. Pentland proposed a method for FDE that utilizes fitting the image into a fractional Brownian function (FBF) and calculating its Fourier's power spectrum $P(f)$ [18]. These two variables are related by the power function and FD estimate D for the image intensity surface could be derived from the exponent of this relation. This method is very accurate, but it requires the image to belong to a class of intensity surfaces that can be modeled using the concept of FBF which is suitable for describing the features of real surfaces within limited range of scales [19]. What is more, its computational complexity is very high as the algorithm involves fast Fourier's transformation (FFT) calculation [17] so its application to, e.g., real-time video processing might cause difficulties due to significant

overheads. Pentland also proposed a fractal surface model providing a surface descriptor applicable to smoothness level assessment, i.e., a difference† between fractal and topological dimensions $D - T \stackrel{\text{def}}{=} r \in [0, 1]$ and a technique for three-dimensional (3-D) shape estimation unifying shaded and textured surfaces [20].

Pentland, like the majority of mentioned authors of publications regarding FDE, utilizes a compendium of 112 grayscale texture photographic images taken under controlled lighting conditions by Phillip Brodatz and published in 1966 [21]. *Brodatz textures* are used as a common reference data set for testing and comparing certain types of image processing algorithms despite their copyright status prohibiting reproduction in publications [22].

Jean-Jacques Gagnepain and Claude Roques-Carmes proposed *reticular cell counting (RCC)* [23] – a method for FDE of statistically (approximately) self-similar objects that are often encountered when processing real images and textures. This approach utilizes counting *space-intensity cells* also called *spicels* that are boxes partitioning 3-D grayscale image intensity space and containing at least one sampling unit representing non-zero image intensity. These spicels are thus defined as countable boxes. It can be shown that, if L is the size of the box corresponding to a specific box scale and N_L is the number of countable boxes for this spicel scale, then:

$$N_L \propto L^{-D}, \tag{1-5}$$

where D is FD estimate that can be calculated as additive inverse of least squares linear fit slope of $N_L(L)$ log–log plot of different box scaling ratios. A noteworthy drawback of RCC approach is that this method cannot accurately assess high FDs as such situations tend to result in high dispersion of samples in Oz axis which in turn causes saturation at a value of about 2.5. On the other hand, its computational complexity is similar to DBC method [17] that makes the time efficiency efficieny of the algorithm very high.

† This is denoted by r by the original author – not to be confused with the same symbol used for different variables used by other autors and excerpted futher. Such usages of the same symbol for different meaning are used occasionally in this literature review and have been exploited as a mean of not altering the notation used in the original pubications. Therefore, if not stated otherwise, a particular method described here and its notation need to be treated in separation from the other ones to avoid notational conflicts. This is caused by the specific character of the literature review and does not apply to further chapters, e.g., a care has been taken to maintain the notation given in part II consistent with the further parts.

Richard F. Voss [24, 25] described the probability $P(m, L)$ that there are m points within a *hypercube* (or a *hypersphere*) of size L centered about an arbitrary point in a set of points. It is normalized in a way that:

$$\forall L. \sum_{m=1}^{N} P(m, L) = 1, \tag{1-6}$$

where N is the number of possible points within the hypercube that in case of 3-D grayscale intensity surface might be a box analogous to that of RCC method. The number of these boxes needed to cover the whole set (image) can be expressed probabilistically as:

$$N_L(L) = M^2 \sum_{i=1}^{N} \frac{1}{m} P(m, L) \propto \sum_{i=1}^{N} \frac{1}{m} P(m, L). \tag{1-7}$$

The combination of eqs. (1-5) and (1-7) allows an estimation of FD by linear regression and mean squared error (MSE) giving a new method that might be briefly referred to as probabilistic RETICULAR CELL COUNTING (PRCC).

As RCC and PRCC method both suffer from the same limitation caused by excessive image intensity variability for higher fractal dimensionality, James M. Keller, Susan S. Chen and Richard M. Crownover [26] have introduced a modification in Voss' method that shifted the saturation level for FD value of about 2.8 at the expense of mildly lengthened computation time [17]. This variant of cell counting approach can be described in short as improved PROBABILISTIC RETICULAR CELL COUNTING (IPRCC) method. It enhances PRCC by applying a linear interpolation to approximate image intensity surface between central points of boxes and their neighbors to create a new surface that is used in subsequent calculations of m on the basis of eq. (1-7). This ensures better nonetheless still limited robustness to discrepancies caused by the quantization effect occurring for irregular yet continuous functions, e.g., fractional Brownian motion (FBM) function [26] introduced by Mandelbrot and John W. Van Ness in 1968 [27] and lately widely used for generating synthetic surfaces as benchmarks for FDE algorithms [28]. Boundary effects of this method as boxes scale decreases and number of nonempty (countable) boxes saturates and become close to limiting value equal to total number of discrete points in the image have been later also studied by IPRCC authors. As a result, in [29]

they present a proof for box size saturation lower bound L_B and its dependency on FD:

$$L_B = \frac{L_{max}}{(M^2)^{1/D}}, \tag{1-8}$$

where L_{max} is the size of a box enclosing the entire fractal image.

Min Long and Fei Peng designed BC method with adaptable box height also referred to as the integer ratio based BC method [30]. They have shown that it has better accuracy for images of arbitrary non-square size. Moreover, their method allows the height of the box at the top of each image array grid block to be adaptable to the maximum and minimum intensity values of that block. This allows some of the common existing BC methods pitfalls to be avoided.

Another approach that has been demonstrated to combine accuracy close to Pentland's FBF-based spectral method with time efficiency even better than Gagnepain and Roques-Carmes' RCC algorithm is the abovementioned DBC introduced by Sarkar and Chaudhuri [17]. They suggested an intensity space partitioning of the image of size $M \times M$ that creates a 2-D array of columns of boxes with each column height equal to total number of gray levels denoted G. The base of each column is a grid of size $s \times s$ image points with scaled-down size $s \in \{2, 3, \ldots, \lfloor \frac{M}{2} \rfloor\} \stackrel{\text{def}}{=} \overline{2, \frac{M}{2}}$ giving scaling ratio $r = \frac{s}{M}$. Numbers of boxes having size of $s \times s \times s'$ points within any column and any row are equal for s' satisfying $\lfloor \frac{G}{s'} \rfloor = \lfloor \frac{M}{s} \rfloor$ condition that ensures $s' \approx \frac{s}{M} \cdot G \stackrel{\text{def}}{=} r \cdot G$ dependency. Boxes in each column are enumerated in a way that bottom box has number one, box above it – number two etc. For a grid of a column that have position (i, j) on array when this column has a minimum and maximum intensity levels falling into boxes k and l, respectively, its grid contribution to total number of counted boxes is:

$$n_r(i, j) = l - k + 1. \tag{1-9}$$

Figure 1.1 (inspired by [17]) illustrates eq. (1-9) for a case where:

$$n_r(i, j) = 3 - 1 + 1 = 3. \tag{1-10}$$

Also, as the example in fig. 1.1 is illustrative and thus simplified to maintain clarity, the image surface presented is smooth, as contrasted to applications exploiting real digitally acquired images that present discrete representations of image surfaces.

DBC method name originates in differential character of the procedure of determining the number of boxes withing a grid that are to be summed up. The mathematical core of this differential approach is expressed by eq. (1-9) and elaborated

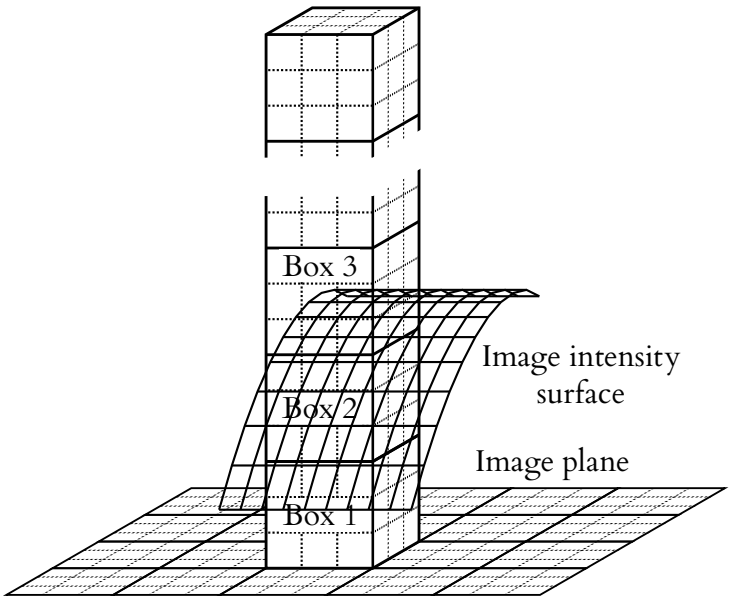


Figure 1.1: Image box partitioning for 3-D interpretation of a grayscale image used for determination of the column's grid contribution n_r to the total number of counted boxes by DBC method (after [17]).

on in detail in [17]. Contributions from all grids in array gathered together give total number of boxes for ratio r:

$$N_r = \sum_{i,j} n_r(i,j). \tag{1-11}$$

This number might be calculated for different ratios (and thus for scaling sizes s). According to Mandelbrot [1], for a self-similar set of FD equal to D that constitutes a distinctive union of N_r its own copies scaled down by ratio r, the following equality is satisfied:

$$N_r = r^{-D}. \tag{1-12}$$

Equation (1-12) implies that when least squares linear fit of $\log(N_r)$ as a function of $\log(r^{-1})$ is found, its slope is the estimate of D. This BC approach is better than RCC, PRCC and IPRCC in a way that it approximates the ϵ-blanket more accurately. The covering of image intensity surface in DBC performs well also for finer resolutions (i.e., smaller scaling ratios) even in case of roughly textured surfaces exhibiting quantization effect on their numerous irregularities [17]. Nonetheless, authors of the described method noticed that in its basic variant it is prone to two limitations that might lower FD estimate value with respect to its anticipated value. The algorithm may fall short of expectations for some specific classes of the textures, i.e., smooth textures with FD values like 2.15 whose in-

tensity level variability range lays within single block number and periodical textures that may even have high FD close to 3, but that are periodical with spatial period of variation causing a situation where biggest differences in image intensity does not fall into single grid area over a vast fraction of the image array. In [17] Sarkar and Chaudhuri also propose modifications offering a workaround to these flaws: a random vertical (in 0z axis) column shifts to improve algorithm performance on smooth surfaces and random horizontal column shifts (in 0x and 0y axes) that increase its accuracy in terms of periodic patterns. The same authors utilized modified DBC algorithm for a study on FD applications in texture classification and segmentation [31] that included utilization of lacunarity and six FD-based features together with other means to improve feature extraction and a comparison of the results with segmentation algorithms exploiting on other popular texture features.

Stéphane Buczkowski et al. [32] proposed a *modified BC method* mitigating two important pitfalls of the general BC approach, i.e., border effect caused by non-integer number of boxes along image border (resulting in border boxes smaller than internal ones) and non-integer number of image pixels along box border (resulting in non-integer values of some scale size values). Sarkar and Chaudhuri's differential variant of BC methodology is especially susceptible to under-and overestimation of the number of boxes and has other disadvantages that have been addressed by Wen-Shiung Chen, Shang-Yuan Yuan and Chih-Ming Hsieh in [33] who subsequently proposed two methods called *shifting* and *scanning DBC* designed to enhance the numerical determination accuracy of FD values.

Jian Li, Qian Du and Caixin Sun proposed further modifications of Sarkar and Chaudhuri's method [28]. Their three changes address: smaller box height r' selection through automatic adjustment based on image smoothness, more accurate box number calculation based on intensity level range d_z within column instead of difference of two boxes numbers based directly on intensity z:

$$n_r(i,j) = \begin{cases} \left\lceil \dfrac{l-k}{r'} \right\rceil & \text{for } r \neq k \text{ or} \\ 1 & \text{for } r = k \end{cases}. \tag{1-13}$$

and *image intensity surface partition* removing gaps between spatially adjacent boxes and thus assuring complete grids coverage of the image and reducing estimation errors. The authors also compare a few variants of their improved DIFFERENTIAL BOX-COUNTING (IDBC) method with the original DBC method and other FDE

methods by the means of FD estimates consistency, regression fit error comparison, FD variation range exploration and image categorization (classification) capabilities for synthetic fBm and real images including video frames with different levels of degradation.

Jibitesh Mishra and Shubha Prasad Pradhan described various methods of FD calculation for grayscale images and present key computational problems involved in DBC method with some experiments and methods results comparison [34]. The article identified problem associated with box height selection in IDBC method. The standard deviation (σ) and mean (μ) of each window of an image are calculated in order to establish negligible height. FD varies within a range determined by IDBC method for different values of α coefficient. For a grayscale image with $\alpha = 3$, its FD is found to exceed 3 which limits the accuracy of IDBC method. Mishra and Pradhan proposed a modified approach with alternate selection of box height and calculation of box number but the theoretical and experimental evaluation of this methodology is relatively narrow as compared to the aforementioned publications regarding DBC improvements.

Ton Dieker described three exact and many approximate methods of fBm signals simulation as well as a set of parameter estimation and testing methods for evaluation of the approximate ones in [35]. Markos Markou and Sameer Singh described a method that might be applicable for such statistical evaluation called *novelty detection* in [36]. It depends on the identification of data or signal unknown for a machine learning system at the time of training set presentation. Work of Markou and Singh is a survey of novelty detection using statistical approaches, and that approach is especially but not exclusively relevant for classifiers utilizing artificial neural networks [36]. Texture classification applications of fBm dimension estimation are further investigated in [37]. Additional applications of fractal-based quantities to image segmentation and contour detection are described in works of another authors, e.g., in [38] describing texture analysis operations on fractal-coded image without the need to decompress it. A description of another four synthesis methods for discrete fBm and their analysis with applications to image processing has been provided by Warren M. Krueger et al. in [39]. Dirk P. Kroese and Zdravko I. Botev included generation of multidimensional fBms, e.g., *fractional Brownian surface* or *field* in [40] that overall is focused on the spatial processes being both Gaussian and Markov. Their work also put an emphasis on spatial point processes and their simulation. Also, problems of constructing spatial processes from the Wiener process and Lévy processes as well as fields generation are concerned. For Brownian surfaces a Cholesky's decomposition method

might be applied as they are realizations of a Gaussian process with a nonstationary covariance function. Better efficiency is assured by Michael L. Stein's method [41]. In this methodology an auxiliary stationary Gaussian process is generated using the approach called circulant embedding. This intermediate process is then altered to deliver the nonstationary Gaussian process generating desired Brownian field.

Antonio Napolitano, Sara Ungania and Vittorio Cannata presented a review of FDE methods for applications in biomedical images [42] including BC algorithm for binary images, DBC algorithm for grayscale images and *hand and dividers method* for operating on the contours of image boundary structures and estimating boundary FD. Authors implemented these methods in MATLAB R2010a and applied them to either well-known fractals or biomedical images and performed comparative analysis of the resulting output data.

David da Silva et al. have exploited binary BOX-COUNTING (BBC) method extended to 3-D variant and using voxel-based boxes to perform FDE on images of artificial tree crowns of 3-D synthetic plant foliage generated using, i.a., iterated function system (IFS) [43].

The presented literature review does not exhaust the topic. Yet, it highlights the most important and notable – in a view of the author – developments in fractal-based descriptors of grayscale images.

1.2.3 Color images

The majority of widely used FDE methods is defined for one-dimensional (1-D) signals or binary images and some of them extend to grayscale images. Mihai Ivanovici and Noël Richard proposed an extension of existing probabilistic algorithm to color images in [44]. They also proposed an extension of the existing probabilistic fractal generation routine that produces color images with desired FD. Their work has been based on Voss' RCC algorithm [24] – for empirical FD calculation – – and *midpoint displacement algorithm* described by Heinz-Otto Peitgen and Dietmar Saupe in [45] – for generation of synthetic data for the purpose of validation, cf. chapter 5.

It can be noticed that the issue of estimating dimensionality and other fractal features of images involving color components is much less covered in literature as compared to their monochrome or grayscale counterparts. While this field of study is not much of interest from the perspective of applications considered in this thesis that focus on grayscale images, it should be noted that the prob-

lem is still to be explored by the scientific community. It is worth mentioning that some aspects of spatial multi-channel fractal analysis are of much higher complexity than of the single-channel one that poses significant challenge for further research and may require construction of new concepts and improved (e.g., possibly more noise-robust) subroutines that would also be applicable to assessment of other classes of signals.

1.2.4 Applications of FDE in biomedical research

FD analysis and estimation is a matter of interest in biomedical research. This subject finds its applications in diagnostic and detection tasks altogether, hence it is also utilized as a diagnostic tool in biomedicine. The applications also include modeling, identification of parameters and planning of experiments as dimensional analysis is studied in general in relation to applications in identification of mathematical models in physics, technology and related sciences [46]. Apart from dimensional analysis and design of experiments, fractal studies remains in strong relationship also with identification theory, algorithmics, numerical experimentation etc. Therefore, in biomedical research, as well as in technology, physics or economy, fractal-focused approach allows for an empirical – or data-driven – description of processes [47]. Thus, it forms a vital diagnostic component that enhances an interdisciplinary insight which may be necessary for analyzing complex problems defined in research area encompassing more than one branch of the traditional fields of science.

The issue of FD analysis remains a subject of investigation in biomedicine, e.g., querying the Google Scholar bibliographic database for a pair of terms "fractal dimension" and 'biomedical' has returned almost 30 thousands of scientific works of scholarly literature at the time of the development of this thesis [48]. In particular, FDE is investigated in context of ophthalmology. In particular, there have been attempts to automate the estimation of tear film surface quality (TFSQ) index to close the gap between the manual and automated tear film quality assessment by means of the high-speed videokeratoscopy technology with applications in a clinical practice [49]. Of special interest, intensely investigated is *the retinal circulation* within arterial and venous systems of retinal vasculature vessel pattern. Within this matter, there have been numerous research publications regarding, i.a., human bronchial tree with focus on embryological development [50], creating semi-automated methods for measurement of retinal vessels and ocular blood

pressure [51], the role of fractal pattern formation in context of angiogenic factors during the development of a human eye [52] as well as embryological and clinical implications of fractal characteristics of branching patterns in retinal vessels [53]. Moreover, fractal dimensional analysis involving spectral-domain optical coherence tomography angiography (OCTA) has been investigated in terms of its diagnostic ability for retinal vascular disease burden of diabetic retinopathy [54]. Another application of fractal analysis to OCTA encompasses the development of a fully automated method of quantification of vessel density and foveal avascular zone (FAZ) area using local FD [55]. Furthermore, the reproducibility of FD has been used as one of parameters used in an attempt to compare seven different OCTA devices in examination of parafoveal microvascular anatomy of superficial and deep capillary plexus in healthy volunteers [56]. Ophthalmologic research has also included 3-D visualization and fractal analysis of cell assortment of mosaic patches in corneas of rat chimeras [57]. Also, a novel approach has been proposed to utilize computational algorithms for fractal analysis of surfaces together with atomic force microscopy (AFM) imagery and to investigate the ultrastructural details of the surface of Bowman's membrane of the human cornea using surface structure data analysis – this has revealed fractal structure in the surface of Bowman's membrane at the nanometer scale [58]. In other branches of ophthalmologic research, retinal microgliosis is one of aspects of an extensive study which has exploited fractal analysis based on, i.a., BC, to determine that overall microglial morphology increases with ramification of the cells [59]. This study has also raised the topic of multifractal analysis and lacunarity for identifying intermediate microglial morphology forms or states. Other analysis has focused on patch clamp recordings of the corneal endothelium to show that the kinetics of its ion channels exhibit fractal characteristics [60]. One important conclusion has been that fractal-based model is more consistent with the experimental data and better in describing conformational dynamics of proteins than models utilizing discrete Markov's states that have often been used in former studies. The applications of fractal analysis to ophthalmology also include pathology, e.g., vascular pathology in retinal vessels has been examined using FD in terms of increased neovascularization near the optic disc that has formed a possible criterion for classifying patients for panretinal laser treatment [61].

2 AIMS AND OBJECTIVES

The main goals of this thesis are:

(1) construction of fractal dimension (FD) estimators for two-dimensional grayscale images that would satisfy the following conditions:
   (1a) numerical efficiency;
   (1b) high dynamic range;
   (1c) consistency with Euclidean (topological) dimension for non-fractal (in a self-similarity sense) sets;

(2) applicability of the newly designed FD estimators to the problem of interferometric imaging of tear film in human subjects.

The objectives of the work are:

(1) investigating FD estimators properties (research in terms of existing solutions);
(2) considering new approaches to FRACTAL DIMENSION estimation (FDE):
   (2a) probabilistic;
   (2b) non-probabilistic.

Part II

METHODOLOGY AND THEORETICAL DEVELOPMENTS

The following part contains an original contribution – a description of the new algorithms to calculate fractal dimension.

3 BOX-COUNTING IN FRACTAL DIMENSION ESTIMATION

3.1 BACKGROUND

A group of methods for fractal dimension (FD) calculation involves numerous variants of the set partitioning (usually grid-based) approach called the *box-counting (BC)* method. It is connected to a more general concept called Hausdorff's FD, but because of its relative simplicity, it is often preferred over the latter in applications related to numerical data analysis. Still, being a fairly abstract paradigm itself, it offers a vast multitude of realizations by different variants, conventions and sub--methods. Accordingly, it is sometimes perceived as an universal concept providing the feasibility and comprehensiveness.

There are different approaches to calculating FD of an image, including binary BOX-COUNTING (BBC) and its differential extensions, including the differential BOX-COUNTING (DBC) algorithm [17, 31], as well as the multitude of other image fractal descriptors [23, 24, 26, 28, 29, 32, 33, 63–67]. However, despite the need for developing a unified approach [7], there is a continuing interest in developing new estimators of FD and other measures of pixel organization within an image [68, 69]. For gray-scale images, the classical BBC FD estimator requires a pre-processing step of transforming the image into binary values, where the result depends on the used threshold [70, 71]. This requirement limits the generality of BBC method to a specifically pre-processed class of input images.

The aim of this thesis has been to develop methods that overcome the limitation of BBC, but also, following the philosophy of Huang, Lorch and Dubes [10] that have the ability of high discriminative power in application in which interferometry is used to non-invasively assess the kinetics of human tear film [72–74].

FD obtained by BC method is called box-counting dimension. It is sometimes also referred to as Minkowski–Bouligand dimension, although this name may be ambiguous, as it is considered by some authors as belonging to a separate FD family that is based on the study of the influence area created by shape dilations using a disc of a varying radius r [75]). The dimension may be formally defined using

This chapter is based on author's coäuthored paper submitted to *IEEE Access* [62].

the limit of box size approaching zero (q.v. definition 2), but in practical applications it is hardly ever calculated directly from this definition. Conversely, it is of common pragmatic usage to use not the dimension value in the formal sense, but rather its estimate based on the linear regression. The measures calculated and discussed in applied and technical sciences, e.g., in digital images processing, tend to be estimated by linear regression from a finite set of data points where each of them corresponds to a separate box partitioning of a different scale. Mathematical formulations regarding this matter are demonstrated, i.a., by eq. (3-5a).

3.1.1 Mathematical description

Equation (3-1) within definition 2 describes BC FD as arising from the logarithmic relationship between the number of boxes covering the examined object and the size of the box. It is worth acknowledgment that the boxes used are typically square--shaped what is also the assumption of further considerations discussed in this work.

Definition 2. BC-based FD:
D is box-counting fractal dimension of a set A when the following limits exist:

$$D \doteq \dim_{\text{box}}(A) = \lim_{\epsilon \to 0^+} -\log_\epsilon N_\epsilon = \lim_{\epsilon \to 0^+} \frac{\log N_\epsilon}{\log \frac{1}{\epsilon}}, \tag{3-1}$$

where the set is covered entirely by boxes of the size ϵ and N_ϵ is the minimal number of boxes required to provide the coverage. □

3.1.2 Numerical estimation

In practical applications, FD obtained by box-counting is often estimated numerically from digital image representation of the object by estimating the slope of the linear regression line calculated for the log–log scale relationship between N_ϵ and ϵ^{-1}. Data points used for regression are calculated for at least few different ϵ values within the suitable range. Specifically, the box sizes are not to be greater than the examined object itself and not to be smaller than the image discretization unit. Often, there would be boundary effects induced by discrete nature of the finite numerical approximation of a theoretical set into image matrix. To mitigate them, some cut-off on both ends of ϵ range might be applied. E.g., for an image of size 128 px × 128 px the smallest box size considered might be 4 px × 4 px or 8 px × 8 px and the biggest 64 px × 64 px.

It may be denoted in matrix form as:

$$\begin{bmatrix} \log N_{\epsilon_1} \\ \log N_{\epsilon_2} \\ \vdots \\ \log N_{\epsilon_k} \\ \vdots \\ \log N_{\epsilon_n} \end{bmatrix} = \begin{bmatrix} \log \epsilon_1^{-1} & 1 \\ \log \epsilon_2^{-1} & 1 \\ \vdots & \vdots \\ \log \epsilon_k^{-1} & 1 \\ \vdots & \vdots \\ \log \epsilon_n^{-1} & 1 \end{bmatrix} \begin{bmatrix} D \\ C \end{bmatrix}, \tag{3-2}$$

where $n \in \mathbb{N}_{>0}$ is the total number of box sizes, i.e., the number of data points to perform linear regression on, C is some constant and D is BC dimension value that is to be estimated for the object.

Equation (3-2) can be written in a more compact form:

$$\mathbf{X} = \mathbf{H}\mathbf{P}, \tag{3-3}$$

where:

$$\mathbf{X} \doteq \begin{bmatrix} \log N_{\epsilon_1} \\ \log N_{\epsilon_2} \\ \vdots \\ \log N_{\epsilon_n} \end{bmatrix}, \tag{3-4a}$$

$$\mathbf{H} \doteq \begin{bmatrix} \log \epsilon_1^{-1} & 1 \\ \log \epsilon_2^{-1} & 1 \\ \vdots & \vdots \\ \log \epsilon_n^{-1} & 1 \end{bmatrix} \tag{3-4b}$$

and:

$$\mathbf{P} \doteq \begin{bmatrix} D \\ C \end{bmatrix}. \tag{3-4c}$$

Now, $\mathbf{P}$ may be estimated as $\hat{\mathbf{P}}$ by applying Moore–Penrose pseudoinverse on $\mathbf{H}$ matrix in eq. (3-3):

$$\hat{\mathbf{P}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{X} \tag{3-5a}$$

and D may be estimated as $\hat{D}$, i.e., the first entry of $\hat{\mathbf{P}}$:

$$\hat{\mathbf{P}} \doteq \begin{bmatrix} \hat{D} \\ \hat{C} \end{bmatrix}. \tag{3-5b}$$

The main problem in all box-counting-based algorithms involves developing the suitable method for determining $N_{\epsilon_k} \doteq N_k$ for a given ϵ_k box size ($k = 1, 2, \ldots, n$). Thus, as a requirement to describe this relation, a function defining N_k is to be given in a form that may be realized in algorithm implementation.

Remark regarding N_ϵ ***definition***

Usually, ϵ value is interpreted as the *size* of the box set (e.g., the side length of a hypercube) such that, as in definition 2, $\epsilon \to 0^+$. Conversely, in case of numerical estimation of BC-based FD, box is usually interpreted as a rectangle or – specifically and typically – a square. Then, the box – when subpixel boxes are not considered – could only be as small as a single pixel. Thus, when box size is expressed as an absolute value being its side length in pixels – as opposed to a relative variant based on the ratio of side lengths of the box and the image – it cannot converge to zero. To distinguish between theoretical set-based box which can be infinitesimally small and its digital image-based finite counterpart, the size of the former is further denoted by ϵ (as in the above considerations), whereas the size of the latter is expressed as ε.

3.2 CONSIDERED METHODS FOR n_ε CALCULATION

Remark regarding the convention and notation

In theoretical descriptions FDs are usually defined for sets that frequently are infinite. In such cases, the boxes used for BC dimension estimation are their subsets that may be defined as, e.g., hyperballs or hypercubes. Also, the patterns used to distribute boxes over the object in order to gather data (sometimes called *scanning plans*) may vary as, e.g., both non-overlapping *fixed grid* scans and overlapping *sliding boxes* are used in different versions of BC algorithm. As algorithms described below are designed to be applied to two-dimensional (2-D) images, the following convention is used – if not explicitly stated otherwise – to provide possibly compact and easier to implement notation:

(1) the object is represented by its 2-D image matrix consisting of square pixels having values in $\{0, 1\}$ in case of binary images or in $[0, 1]$ for grayscale images;

(2) the sizes of the images and objects within them, e.g., boxes, are expressed in pixels (px) and the name or symbol of this unit are omitted throughout the work to increase clarity of the formulas;

(3) the image is a square matrix of dimensions $l \times l$ being a non-negative integer powers of two, i.e., $l \doteq 2^m$ ($m \in \mathbb{N}_0$), where m is the image side length binary exponent;

(4) the box having (absolute, i.e., pixel-based) dimensions $\varepsilon \times \varepsilon$ is a square submatrix of the image matrix. Note that a geometrical object is defined in an E-D space [45] and in our case, $E = 2$ as we interpret 3-rd parameter of the image, i.e., its intensity, not as a dimension but the incidence probability. Note also that in this work the box size is changed in steps enumerated by k (starting from $k = 0$) and in each step its value is fixed for the whole image: thus, it would be denoted as ε_k, but for the purpose of notation simplicity, this is simplified to unindexed symbol ε; nevertheless, ε value still explicitly depends on k, q.v. item (5), until it is stated otherwise;

(5) ε can take its value from the following set:

$$\varepsilon \in \{2^0, 2^1, \ldots, 2^k, \ldots, 2^m\}$$

and for k-th division of the box grid, its square boxes have their sides of lengths $\varepsilon \doteq 2^k$ ($k \in \{0, 1, \ldots, m\}$), including the 0-th step for a single box encompassing a single pixel; note that due to the reasons described in section 3.1.2, a few smallest as well as a few greatest elements might be omitted, e.g., $k \in \{3, 4, \ldots, m - 1\}$ may be chosen as the range of the box grid divisions;

(6) box pattern is a fixed non-overlapping grid constructed by the iterative box merging in such way that for any division forming a new step, every new box is constituting by four boxes of previous stage that are placed in its quarters (until further merging is impossible due to image size or a cut-off threshold is reached) with 0-th stage's box size (that would be denoted as ε_0 in the less compact notation) being equal to one pixel. Thus, for a square image that is considered in this thesis, the 1-st stage's box size (correspondingly, ε_1) is $2\,\text{px} \times 2\,\text{px}$, whereas the last stage ($k = m$) involves a single box encompassing the whole image;

(7) because in this thesis there is no need to address the individual pixels or boxes by both width and height of their location and for a specified, k-th division stage, boxes within the image can be unambiguously addressed

as well as pixels within boxes (or also within the image), if not stated otherwise, the continuous enumeration (of pixels within the box or boxes within the image) is used, with left-to-right and top-to-bottom convention starting from the top left entries of box matrix or elements of box grid (while other assignment might be chosen as well as long as it will be unambiguous and complete); e.g., for $i \leqslant \frac{l}{\varepsilon} \doteq \frac{2^m}{2^k} = 2^{m-k}$ and $j \leqslant \varepsilon \doteq 2^k$, $x_{i,j,k}$ denotes value of j-th pixel from the left in the top row of i-th box from the left in the top box row of the image for k-th stage of image box grid division.

By definition, N_ε is derived from the image $\mathbf{A}$ as a function of the size ε of the box within the grid, i.e., $N_\varepsilon \doteq f(\mathbf{A}, \varepsilon)$ and for k-th box grid division involving $\varepsilon = \varepsilon_k$ this may be denoted as $N_k \doteq N_{\varepsilon_k}$. Nonetheless, different methods require separate definitions of f function.

3.2.1 Basic convention of symbolic notation

The following symbols – also illustrated in fig. 3.1 – will be used:

(1) for primary (formal) notation:

- $\mathbf{A}$ — digital image interpreted as a matrix of size $l \times l$ containing the numerical representation (finite approximation) of the object to be examined terms of BC dimension estimation:
  - l, $L \doteq l^2$ — dimensions (size) of image $\mathbf{A}$: height equal to width and the total number of pixels, respectively;
- $\mathbf{B}_{i,k}$ $(i = 1, 2, \ldots, c_k)$ — i-th box having the size $\varepsilon \doteq 2^k$ within the box grid of the image matrix $\mathbf{A}$, i.e., a submatrix of $\mathbf{A}$ of size $\varepsilon \times \varepsilon$ constructed by iterative merging of matrix quarters described above:
  - $c_k \doteq (\frac{l}{\varepsilon})^2 = (\frac{l}{2^k})^2 = (\frac{2^m}{2^k})^2 = 2^{2(m-k)}$ — total number of ε-size boxes required to constitute grid coverage of the image $\mathbf{A}$; this includes boxes that do not include any non-zero pixel, i.e., the parts of image grid partitioning that do not overlay the object area and in turn are not counted into N_ε value; thus, $0 \leqslant N_\varepsilon \leqslant c_k$;
- $x_{i,j,k}$ $(i = 1, 2, \ldots,\ c_k = 2^{2(m-k)},\ j = 1, 2, \ldots,\ \varepsilon^2 = 2^{2k})$ — j-th pixel within i-th box $(\mathbf{B}_{i,k})$ of the grid of ε-sized boxes for k-th box grid division – being the j-th entry in the vectorized box matrix $\mathrm{vec}(\mathbf{B}_{i,k})$:

$$(x_{i,1,k}, x_{i,2,k}, \ldots, x_{i,j,k}, \ldots, x_{i,\varepsilon^2,k})^T \doteq \mathrm{vec}(\mathbf{B}_{i,k}) \in [0, 1]^{\varepsilon \cdot \varepsilon}, \quad (3\text{-}6)$$

where $\mathrm{vec}(\cdot)$ denotes the matrix vectorization operator (used according to the left-to-right and top-to-bottom – or other – order convention, as it is mentioned above);

(2) for equivalent downsampling-based notation†:

- $\mathbf{A}_k$ — digital image obtained from $\mathbf{A}$ by downsampling (with initial matrix $\mathbf{A}_0 \doteq \mathbf{A}$) – a matrix of size $l_k \times l_k$ in which, for a specific ε value, one pixel represents the value obtained from one ε-sized box from $\mathbf{A}$ matrix of the primary notation that is to be counted into N_k during BC procedure:
  - l_k, $L_k \doteq l_k^2 \doteq c_k$ — dimensions (size) of image $\mathbf{A}_k$: height equal to width and total number of pixels (i.e., of corresponding ε-sized box grid elements in $\mathbf{A}$), respectively;
- $x_{i,k}$ $(i = 1, 2, \ldots, L_k)$ — i-th pixel of $\mathbf{A}_k$ (for k-th box grid division); image $\mathbf{A}$ needs to be downsampled in such a way that the value of $x_{i,k}$ pixel of $\mathbf{A}_k$ would be the same as the contribution of the corresponding i-th ε-sized box ($\mathbf{B}_{i,k}$) of $\mathbf{A}$ to N_k.

† Useful for implementation of the recursive image downsampling procedure that increases the computational efficiency of BBC algorithm and some of weighted grayscale BOX-COUNTING (W--GBC) algorithm variants, cf. section 3.2.4.

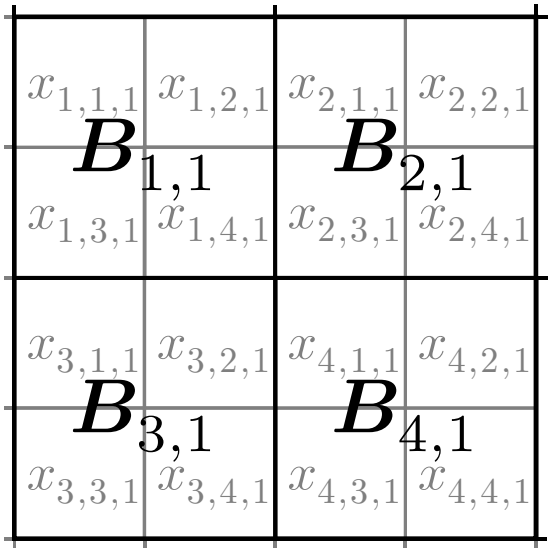


Figure 3.1: Notation used to enumerate the boxes and pixels within them for the second possible image $\mathbf{A}$ box partitioning (i.e., $k = 1$) of $m + 1 = 3$ possible box partitions ($k \in \{0, 1, 2\}$). The total number of boxes within grid is $c_k \doteq 2^{2(m-k)} = 2^{2\cdot(2-1)} = 4$ and the total number of pixels within the image is $L \doteq l^2 = 4^2 = 16$.

3.2.2 Binary images

Basic box-counting algorithm can be applied only to binary images as it regards only the presence of the object (or lack of it) within a given box and does not consider its intensity (or weight, probability, etc.). Therefore, only a binary information is needed and the grayscale data may be exploited only during the pre-processing stage (by setting global or local binarization threshold), but otherwise must be discarded before applying the algorithm.

(Classical) binary box-counting

This is a basic BC algorithm that involves grid partitioning of the image. Counted are these box grid elements that include non-zero pixels. They are boxes overlaying object area, i.e., these enclosing at least one pixel of the value of one (that before the local or global binarisation procedure had value greater than or equal to some threshold value $t \in (0, 1]$). The input image needs to be already binarized and background pixels are assumed to have zero value contrary to foreground object pixels having the value of one. Particularly, when the image is already of the aforementioned zero–one class, $t = 1$ might be chosen for the binarisation procedure preceding the application of eqs. (3-7) and (3-29) given below. On the other side, in case of the grayscale images extension and when the binarisation procedure

is considered to be the part of the algorithm, the infinite amount of possibilities of threshold t choice forms a family of BBC dimensions having its members being determined by the threshold kind (local or global) and value.

EDGE BINARY BOX-COUNTING Images often undergo pre-processing procedure before being subject to the actual image processing algorithm. If two routines of image analysis differ only in subroutines invoked during the pre-processing stage, they are technically the same algorithm despite the fact that both their outputs and efficiency in accomplishing their task of interest could differ dramatically. Thus, from the practical point of view they may be classified as two separate approaches or rather two variants of the same basic technique. If one of them involves only the necessary pre-processing procedure, e.g., common for the whole branch of such algorithm, and the second includes also an extra preparing step of numerical manipulations on the input data, the latter may be practically described as a variation derived from the reference method, but still belonging to very same type of methods. This is a matter of nomenclatural convention and depends on the criteria chosen to distinct separate algorithms. Nevertheless, when a pre-processing may significantly alter the input, but at the same time it is suspected that omitting it may render the whole method inutile, it is justified to test both the basic as the derived variant. Then, some form of naming and classification needs to be chosen to maintain comprehensibility. This is the case for edge BINARY BOX-COUNTING (EBBC). It is technically BBC technique of FRACTAL DIMENSION estimation (FDE) that is calculated by the original binary algorithm, but it operates on the binary map of edges detected in the original input image instead of processing the (binarised) input data itself. It has been introduced and used along with BBC due to the character of BBC approach in context of the specific character of the data used druing the research. BC mesures the dimensionality of the geometric structures and shapes present in the input data. They may include forms resembling, i.a., flat surfaces as well as fine thin lines, and in typical cases, the variations in the structure of the former would cause the result to oscillate about the value of two, while for the latter the value of reference is expected to be closer to one. In this spirit, lateral shearing interferometry (LSI) data and its inherently anisotropic nature of fringes in a variety of stripe- of band-like forms, has been the motivation behind the introduction of EBBC. As BBC is restricted to binary images, a typical case of interferometric image of the class involved in this research, is binarised to a set of separate long objects having the width of over a dozen of pixels, cf., e.g., fig. 7.4. The useful information about the surface kinetics may be encoded not only in the area

of these objects, but also in the subtle spatial and temporal changes of their borders. Therefore, it might be of use to focus not on 2-D variations in the binary fringe pattern, but only on one-dimensional (1-D) wrinkles that appear on the very borders of the fringes. Restricting to one featured direction may be consistent with the lateral and unidirectional form of the acquired images. The simplest way to achieve it is to extract or detect the edges of the image to be processed, and feed the obtained map of edges into the proper algorithm. Whether the straightforward or the edge-based approach would yield better result in terms of data classification, is a question that has no clear a priori answer before relevant experiments are conducted. This is caused mostly due to the obscure, dispersed (or even sometimes dissipated) and thus volatile nature of the exact feature of the interferograms that contains the most useful information, as simple phase-based approach often fails for heavily disrupted images, eg, of tear film on the contact lenses. Consequently, it has been decided to investigate both the basic procedure and the one involving additional pre-processing step of edge detection. In this work, the former is referred to simply to as BBC, while the latter – as EBBC.

PRIMARY NOTATION This convention is suitable for formal theoretical purposes and applies to *both* BBC approach to FDE and EBBC fractal-based descriptor of fractal features in the image:

$$N_k(\mathbf{A}) \doteq \sum_{i=1}^{c_k} \operatorname{any}(\mathbf{B}_{i,k}), \tag{3-7}$$

where matrix operator $\operatorname{any}(\cdot)$ returns zero for a null matrix and one otherwise, i.e., when matrix contains at least one non-zero entry. Specifically, if matrix $\mathbf{A}$ represents an image where background pixels have zero value, $\operatorname{any}(\mathbf{A})$ would return one if and only if the image contains at least one object pixel.

3.2.3 Grayscale images

Developing a method that utilizes the grayscale in its core and not only during the pre-processing stage may be beneficial. Such approach may extract more useful information from the image that otherwise would be permanently lost due to binarization. This additional information may be useful at further stages, e.g., for improving an image classifier constructed upon the estimated FD values.

One of the popular approaches to estimating FD from grayscale images is the DBC algorithm of Sarkar and Chaudhuri [17, 31].

Probabilistic grayscale box-counting

This variant arises upon the image interpretation stating that its intensity represents the local irradiance (or, more specifically, charge-coupled device (CCD)-photoactive flux†) of the light-wave. This quantity directly corresponds to the incidence probability of a photon path and imaging unit (pixel) area of the image plane. Such probabilistic approach may be further extended to the utilization of probability-based image descriptors, e.g., information (Shannon's) entropy [76, 77], which is elaborated below.

PRIMARY NOTATION This convention for description of *probabilistic* approach to grayscale BOX-COUNTING (GBC) method, further referred to in an abbreviated manner as P-GBC, is mostly suitable for formal theoretical purposes:

$$N_k(\boldsymbol{A}) \doteq \sum_{i=1}^{c_k} H_r(\mathbf{B}_{i,k}) \qquad (r \in \mathbb{R}_{>0}), \tag{3-8}$$

where $H_r(\mathbf{B}_{i,k})$ denotes image entropy of the i-th ε_k-sized box employing base-r logarithm and $H_r(\cdot) \doteq H_{r,\,\mathrm{img.}}(\cdot)$ (resulting in $N_k \in [0, 2c_k \log_r \varepsilon]$). Alternatively, $H_r(\cdot) \doteq H_{r,\,\mathrm{int.}}(\cdot)$ (resulting in $N_k \in [0, c_k \log_r U]$), depending on the probability calculation space defining the internal sub-variant of the algorithm: image--based- or intensity-based, respectively.

IMAGE-PROBABILITY-SPACE-BASED ENTROPY Regarding the case of probabilistic GRAYSCALE BOX-COUNTING (P-GBC) method variant, the *image* probability space is utilized in which the probability is directly proportional to pixel values. Thence, the name of this variant is abbreviated as P-GBC-IMG. Here, image en-

† CCD-photoactive flux may be interpreted as the measure of the light power that is converted to brightness signal by the image sensor after the process of electrical charge generation and transfer. This power is almost always lower than the total light power physically absorbed by the sensor due to its specific spectral sensitivity described by the curve of the relative detection efficiency. This effect is responsible for decreasing the wavelenght-related brightness of the acquired image. In that sense, CCD-photoactive flux is CCD's *analogue* of the luminous flux (also called luminous or visible power). The latter is used to weight the received radiant flux wavelength-dependend power components to adjust them to light sensitivity of the average (or – more precisely – standardized) human eye. This sensitivity also varies in relation to the wavelength. SI system defines separate unit for the luminous flux – the lumen (lm) – but not for CCD-photoactive flux. Nevertheless, both may be equivalently expressed in the units of power, e.g. watts (W), as their values describe the amount of light power converted to (percieved or acquired) visual signal.

tropy and photon–pixel incidence probability definitions are described by eq. (3-9). Providing that $\forall j \in \overline{1, \varepsilon^2}.\, p_{\text{img.}}(x_{i,j,k}) \neq 0$ †:

$$H_{r,\,\text{img.}}(\mathbf{B}_{i,k}) \doteq \sum_{j=1}^{\varepsilon^2} p_{\text{img.}}(x_{i,j,k}) \log_r \frac{1}{p_{\text{img.}}(x_{i,j,k})} \in [0, 2 \log_r \varepsilon] \qquad (r \in \mathbb{R}_{>0}), \tag{3-9a}$$

where typically $r = 2$ (or, e.g., $r = e$, $r = 10$ etc.) and $p_{\text{img.}}(\,\cdot\,)$ is *local* (box-wise) photon–pixel incidence probability:

$$p_{\text{img.}}(x_{i,j,k}) \doteq \frac{x_{i,j,k}}{\sum_{t=1}^{\varepsilon^2} x_{i,t,k}} \in [0, 1]. \tag{3-9b}$$

INTENSITY-PROBABILITY-SPACE-BASED ENTROPY In case of probability calculated in *intensity* probability space, a variant of P-GBC abbreviated as P-GBC-INT, the probability is proportional to the numerousness of given range of pixel values. Thus, quantities defined in eq. (3-9) are to be exchanged to these stated by eq. (3-10), i.e., image entropy and histogram bin fitting probability. Analogously to eq. (3-9a), providing that $\forall u \in \overline{1, U}.\, p_{\text{int.}}(b_u, i, k) \neq 0$:

$$H_{r,\,\text{int.}}(\mathbf{B}_{i,k}) \doteq \sum_{u=1}^{U} p_{\text{int.}}(b_u, i, k) \log_r \frac{1}{p_{\text{int.}}(b_u, i, k)} \in [0, \log_r U] \qquad (r \in \mathbb{R}_{>0}), \tag{3-10a}$$

where b_u ($u \in \{1, 2, \ldots, U\},\ U \in \mathbb{N}_{>0}$) is u-th bin (interval) of image histogram out of overall U intervals uniformly distributed in image intensity probability space, while $p_{\text{int.}}(\,\cdot, \cdot, \cdot\,)$ is *local* (box-wise) histogram bin probability:

$$p_{\text{int.}}(b_u, i, k) \doteq \frac{\sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x : x \in b_u\}}(x_{i,t,k})}{\varepsilon^2} \in [0, 1]. \tag{3-10b}$$

Weighted grayscale box-counting

This variant utilizes various manners of averaging (including a special case of the maximum-based average) of the pixel intensity values in a given box. The averaged box intensity value constitutes its *weight* being counted as contribution of this box

† Alternatively to this provision – for the purpose of definitions of image entropy presented in this thesis – omission of any addends containing logarithms of such probabilities (or their inverses) could be applied to the summation within these definitions. This may be achieved in actual implementation by, e.g., substituting zero in place of $\log p_{\,\cdot}(\,\cdot\,)$ and $\log \frac{1}{p_{\,\cdot}(\,\cdot\,)}$ in cases where $p_{\,\cdot}(\,\cdot\,) = 0$.

to the cumulative N_ε value. Such approach originates in the statement that the significance of a box encompassing a part of the object may depend greatly on the most intense entry within that box. This value may be obtained using $\max(\cdot)$ operator that is a special case of the generalized mean (also called power or Hölder mean) operator defined in eq. (3-12a). Furthermore, conclusion may be drawn that the definition of this category of BC-based FD estimators may be generalized to utilize other power mean cases in its sub-variants.

PRIMARY NOTATION This convention is suitable for formal theoretical purposes:

$$N_k \doteq \sum_{i=1}^{c_k} \mu_p(\mathbf{B}_{i,k}) \in [0, c_k] \qquad (p \in \mathbb{R} \cup \{\pm\infty\}), \tag{3-11}$$

where $\mu_p(\mathbf{B}_{i,k})$ denotes the Hölder mean with exponent p of the i-th ε-sized box:

$$\mu_p(\mathbf{B}_{i,k}) \doteq \begin{cases} \lim\limits_{p\to-\infty} m_p(\mathbf{B}_{i,k}) = \min\limits_{j=1,2,\ldots,\varepsilon^2}\{x_{i,j,k}\} & \text{if } p = -\infty \\ m_p(\mathbf{B}_{i,k}) = \left(\dfrac{1}{\varepsilon^2}\sum\limits_{j=1}^{\varepsilon^2} x_{i,j,k}^p\right)^{\frac{1}{p}} & \text{if } p \in \mathbb{R}_{\neq 0} \\ \lim\limits_{p\to 0} m_p(\mathbf{B}_{i,k}) = \left(\prod\limits_{j=1}^{\varepsilon^2} x_{i,j,k}\right)^{\frac{1}{\varepsilon^2}} & \text{if } p = 0 \\ \lim\limits_{p\to+\infty} m_p(\mathbf{B}_{i,k}) = \max\limits_{j=1,2,\ldots,\varepsilon^2}\{x_{i,j,k}\} & \text{if } p = +\infty \end{cases} \in [0, 1], \tag{3-12a}$$

and:

$$m_p(\mathbf{B}_{i,k}) \doteq \left(\frac{1}{\varepsilon^2}\sum_{j=1}^{\varepsilon^2} x_{i,j,k}^p\right)^{\frac{1}{p}} \quad \text{if } p \in \mathbb{R}_{\neq 0}\,. \tag{3-12b}$$

where – in a typical case – the choice for p may be such that $p = +\infty$ (when generalized mean returns maximum intensity value within box) – this variant has been chosen to be used throughout the experiments as it tended to yield, in the majority of cases, the best results during the preliminary research. Note that for binary images, W-GBC reduces to BBC in case of $p = +\infty$. Alternatively, p might be chosen as $p = 2$ (quadratic mean related to energy-like image intensity distribution parameters), $p = 3$ (cubic mean related to asymmetry-like image intensity distribution parameters) or other – possibly non-integer – values.

3.2.4 Downsampling-based definitions

The downsampling-based notation convention is suitable for practical implementation of the algorithms as its application may improve computational complexity by reducing the number of arithmetical operations to be performed for some of them. This may be achieved by performing calculations for subsequent steps k of box sizes ($\varepsilon \doteq \varepsilon_k$) not by directly processing the input image $\boldsymbol{A}$, but rather by exploiting the data already obtained in previous steps. Then, the main analyzed object is an image $\boldsymbol{A}_{k-1}$ ($\boldsymbol{A}_0 \doteq \boldsymbol{A}$) that for $k \geqslant 2$ counts $2^{2(k-1)}$ times fewer pixels to process than the original image matrix $\boldsymbol{A}$ and thus may be analyzed in adequately shorter time. Such approach is called *recursive downsampling technique* and is further elaborated on below.

For all BC variants:

$$N_k(\boldsymbol{A}_k) \doteq \sum_{i=1}^{L_k} x_{i,k} \doteq \sum_{i=1}^{L_k} g(\boldsymbol{B}_{i,k}) \doteq N_k(\boldsymbol{A}), \tag{3-13}$$

where $L_k \doteq l_k^2 \doteq c_k$ is number of pixels $x_{i,k} \doteq g(\boldsymbol{B}_{i,k})$ within the image $\boldsymbol{A}_k$ equal to the number of boxes $\boldsymbol{B}_{i,k}$ within the image $\boldsymbol{A}$ and $g(\cdot)$ is (box-wise) downsampling function operating on boxes $\boldsymbol{B}_{i,k}$ of image $\boldsymbol{A}$ to generate downsampled image $\boldsymbol{A}_k$. When the recursive downsampling variant, described below, is to be applied, $g(\cdot)$ operates on boxes of $\boldsymbol{A}_{k-1}$ instead of boxes of $\boldsymbol{A}$.

Importance

The motivation for downsampling-based definitions has been the search for a compact design of the mathematical apparatus used, in a sense that it shall not incorporate the details of the downsampling procedure itself, as that would substantially complicate the formal notation. On the other hand, the equivalent definitions based on downsampling process are useful from a practical point of view, as it is demonstrated below.

Recursive downsampling technique

NOTATION The image matrix division into boxes might be started from the single-pixel boxes which are then recursively merged into major (larger) boxes further on, as it is described in section 3.2.1. If that is the case, sometimes $\boldsymbol{A}_{k+1}$ image matrix can be calculated by downsampling not only directly from $\boldsymbol{A}$, but also from $\boldsymbol{A}_k$, depending on the properties of internal downsampling function.

Lemma α. Unambiguous boxes to sub-boxes mapping:
If for a k-th step that is feasible ($k \in \mathbb{N}_0,\ 1 \leqslant k \leqslant m \in \mathbb{N}_0$) the quarters merging method (described above) applies:

$$\mathbf{B}_{i,k} \doteq \begin{bmatrix} \mathbf{B}_{i_1(i),k-1} & \mathbf{B}_{i_3(i),k-1} \\ \mathbf{B}_{i_2(i),k-1} & \mathbf{B}_{i_4(i),k-1} \end{bmatrix}, \tag{3-14}$$

then the set of sub-boxes indices from step k:

$$\{ i_1(i), i_2(i), i_3(i), i_4(i) : i = 1, 2, \ldots, c_k \},$$

where $i_1(i)$, $i_2(i)$, $i_3(i)$ and $i_4(i)$ denote indices of four sub-boxes of box $\mathbf{B}_{i,k}$, exhibits one-to-one correspondence to the set of basic boxes for step $k-1$:

$$\{ i : i = 1, 2, \ldots, b_{k-1} \}$$

and both sets may be partitioned into unique families of their four-element subsets of which each family bijectively corresponds to the set of all boxes $\mathbf{B}_{i,k}$ from k-th step. □

Proof:
By definition, every box $\mathbf{B}_{i,k}$ of the image A to be used in step k is a square matrix that might be fully reconstructed from its four non-overlapping sub-matrices (that also are square). Moreover, these sub-boxes constitute the basic boxes for the previous step $(k-1)$. As the sub-boxes set from current step and matrices set from previous step are not multisets and are the same set, every element of the former corresponds unambiguously to an element in the latter and likewise are their indices. ■

Furthermore, continuous enumeration of boxes within grid fixed upon image using left-to-right and top-to-bottom convention possesses useful property. Specifically, it may be shown that explicit formula exists for k-th step to transform indices

of four sub-boxes of $\mathbf{B}_{i,k}$ $(i = 1, 2, \ldots, c_k)$ into indices of their box counterparts from $k-1$-th step:

$$\begin{aligned}
\underset{k-1}{\mathrm{ind}}(i_1(i)) &= 4l_k((i-1) \div l_k) + 2(i - ((i-1) \div l_k)l_k) - 1 = \\
&= 2^{m-k+1}((i-1) \div 2^{m-k}) + 2i + 1 \\
\underset{k-1}{\mathrm{ind}}(i_2(i)) &= \underset{k-1}{\mathrm{ind}}(i_1(i)) + 1 = \\
&= 2^{m-k+1}((i-1) \div 2^{m-k}) + 2i \\
\underset{k-1}{\mathrm{ind}}(i_3(i)) &= \underset{k-1}{\mathrm{ind}}(i_1(i)) + l_{k-1} = \\
&= 2^{m-k+1}((i-1) \div 2^{m-k} + 1) + 2i - 1 \\
\underset{k-1}{\mathrm{ind}}(i_4(i)) &= \underset{k-1}{\mathrm{ind}}(i_3(i)) + 1 = \\
&= 2^{m-k+1}((i-1) \div 2^{m-k} + 1) + 2i
\end{aligned} \quad , \tag{3-15}$$

where $\underset{k-1}{\mathrm{ind}}(\cdot)$ denotes index of a box from previous $(k-1)$-th step which box corresponds to a sub-box in current k-th step and $\div$ denotes integer division operator.

Theorem 1. Downsampling and recursive idempotence:
The necessary and sufficient condition for the recursive downsampling to be employed is the image downsampling function $g(\cdot)$ having the property of being recursively idempotent, in the sense that for four square matrices $\mathbf{B}_{i_s,k}$ $(s \in \{1, 2, 3, 4\})$ of identical size:

$$g(\begin{bmatrix} \mathbf{B}_{i_1,k} & \mathbf{B}_{i_2,k} \\ \mathbf{B}_{i_3,k} & \mathbf{B}_{i_4,k} \end{bmatrix}) \equiv g(\begin{bmatrix} g(\mathbf{B}_{i_1,k}) & g(\mathbf{B}_{i_3,k}) \\ g(\mathbf{B}_{i_2,k}) & g(\mathbf{B}_{i_4,k}) \end{bmatrix}). \tag{3-16}$$

Proof:
For a feasible step $k \geqslant 1$, there is, by definition, q.v. eq. (3-13):

$$\begin{aligned}
N_k &\doteq \sum_{i=1}^{L_k} x_{i,k}, \\
x_{i,k} &\doteq g(\mathbf{B}_{i,k}).
\end{aligned} \tag{3-17}$$

Using quarters merging method there is:

$$g(\mathbf{B}_{i,k}) = g(\begin{bmatrix} \mathbf{B}_{i_1(i),k-1} & \mathbf{B}_{i_3(i),k-1} \\ \mathbf{B}_{i_2(i),k-1} & \mathbf{B}_{i_4(i),k-1} \end{bmatrix}). \tag{3-18}$$

The assumption of recursive idempotence of downsampling function $g(\cdot)$ implies that:

$$g(\begin{bmatrix} \mathbf{B}_{i_1(i),k-1} & \mathbf{B}_{i_3(i),k-1} \\ \mathbf{B}_{i_2(i),k-1} & \mathbf{B}_{i_4(i),k-1} \end{bmatrix}) \equiv g(\begin{bmatrix} g(\mathbf{B}_{i_1(i),k-1}) & g(\mathbf{B}_{i_3(i),k-1}) \\ g(\mathbf{B}_{i_2(i),k-1}) & g(\mathbf{B}_{i_4(i),k-1}) \end{bmatrix}). \tag{3-19}$$

By definition again:

$$g(\begin{bmatrix} g(\mathbf{B}_{i_1(i),k-1}) & g(\mathbf{B}_{i_3(i),k-1}) \\ g(\mathbf{B}_{i_2(i),k-1}) & g(\mathbf{B}_{i_4(i),k-1}) \end{bmatrix}) \equiv g(\begin{bmatrix} x_{i_1(i),k-1} & x_{i_3(i),k-1} \\ x_{i_2(i),k-1} & x_{i_4(i),k-1} \end{bmatrix}), \tag{3-20}$$

where $x_{\cdot,k-1}$ belongs to $\mathbf{A}_{k-1}$, i.e., the matrix downsampled in previous $(k-1)$-th step.

From eqs. (3-17) to (3-20):

$$x_{i,k} = g(\begin{bmatrix} x_{i_1(i),k-1} & x_{i_3(i),k-1} \\ x_{i_2(i),k-1} & x_{i_4(i),k-1} \end{bmatrix}). \tag{3-21}$$

Lemma α implies that $i_1(i)$, $i_2(i)$, $i_3(i)$ and $i_4(i)$ depend only on i, k and m. For a given input image $\mathbf{A}$, its side length binary exponent m is fixed. Additionally, only FDE procedures that utilize single downsampling function $g(\cdot)$ are considered. Therefore, *for a given image and specific FDE method*, any entry $x_{i,k}$ of downsampled matrix $\mathbf{A}_k$ can be derived only on the basis of the elements of $\mathbf{A}_{k-1}$ and current iteration number k. In this context, also k may be considered as redundant, because it can be inferred directly from $\mathbf{A}_{k-1}$ side length l_k and m, as $l_{k-1} \doteq 2^{m-k+1}$.

Thus, matrix $\mathbf{A}_k$ can be reconstructed in such a recursive way based solely upon $\mathbf{A}_{k-1}$:

$$\mathbf{A}_k = \mathbf{h}(\mathbf{A}_{k-1}), \tag{3-22}$$

where $\mathbf{h}(\cdot)$ is some matrix function.

The number of boxes counted in k^{th} step can be calculated directly from downsampled image matrix $\mathbf{A}_k$, q.v. eq. (3-17). This matrix can be calculated in a recursive manner, i.e., by exploiting the result ($\mathbf{A}_{k-1}$) of the previous step alone: without referring to the input image $\mathbf{A}$, q.v. eq. (3-22), if and only if the downsampling function is recursively idempotent, q.v. eq. (3-19). Therefore, recursive idempotence of the downsampling function is the necessary and sufficient condition for utilization of the recursive downsampling technique. ∎

Example 1. Recursive downsampling function:
This may be illustrated by the simple example of $g \doteq \max$:

$$\max\left(\begin{bmatrix} \mathbf{B}_{i_1,k} & \mathbf{B}_{i_2,k} \\ \mathbf{B}_{i_3,k} & \mathbf{B}_{i_4,k} \end{bmatrix}\right) \equiv \max\left(\begin{bmatrix} \max(\mathbf{B}_{i_1,k}) & \max(\mathbf{B}_{i_3,k}) \\ \max(\mathbf{B}_{i_2,k}) & \max(\mathbf{B}_{i_4,k}) \end{bmatrix}\right). \tag{3-23}$$

□

In a case of downsampling procedure exploiting the recursive idempotence property, the total number of arithmetic operations to be performed is significantly decreased and thus are the processor load and computation time. The main cause behind it are the very few operations needed for the final steps with larger k values (as k gets closer to m for subsequent downsampling operations) effecting in few, yet large, boxes and low resolution downsampled images. Similarly, the amount of memory needed for processing is lower in final stages, nevertheless this does not improve the global memory complexity of the algorithm which is predominantly determined by the first stage as it involves the largest image matrices.

Function $\mathrm{any}(\cdot)$, q.v. eqs. (3-7) and (3-29), together with power mean function $\mu_p(\cdot)$, q.v. eqs. (3-12a) and (3-32), for $p \in \{-\infty, 0, 1, +\infty\}$ both exhibit the recursive idempotence. Conversely, entropy function $H_r(\cdot)$, q.v. eqs. (3-9a), (3-10a) and (3-31), does not posses this attribute. Thus, image downscaling routine might be performed in an efficient recursive manner for BBC algorithm and for four types of W-GBC algorithm, i.e., minimum-and maximum-based, along with these involving geometric and arithmetic means.

Figure 3.2 presents an illustrative instance of recursive downsampling technique for 8 px × 8 px image with 2 px × 2 px and 4 px × 4 px boxes. Here, this technique allows for reduction of matrix elements number to be processed by $\max(\cdot)$ operator of W-GBC algorithm from 128 to 80, i.e., by 37.5%. The general matter of computational overhead reduction in recursive downsampling is further evaluated in the paragraph below.

PROCESSING OVERHEAD REDUCTION IN RECURSIVE DOWNSAMPLING Assuming a complete review of the image matrix during N_k calculation and no cut-offs of extreme k values, the total number $n_1(m)$ of matrix entries to be iterated through without recursive downsampling technique is the number of entries in input image matrix: $L \doteq l^2 = 2^{2m}$ times the number of iterations: $m + 1$ (as k is iterated from 0 to m):

$$n_1(m) = 2^{2m}(m+1) = 4^m(m+1); \tag{3-24a}$$

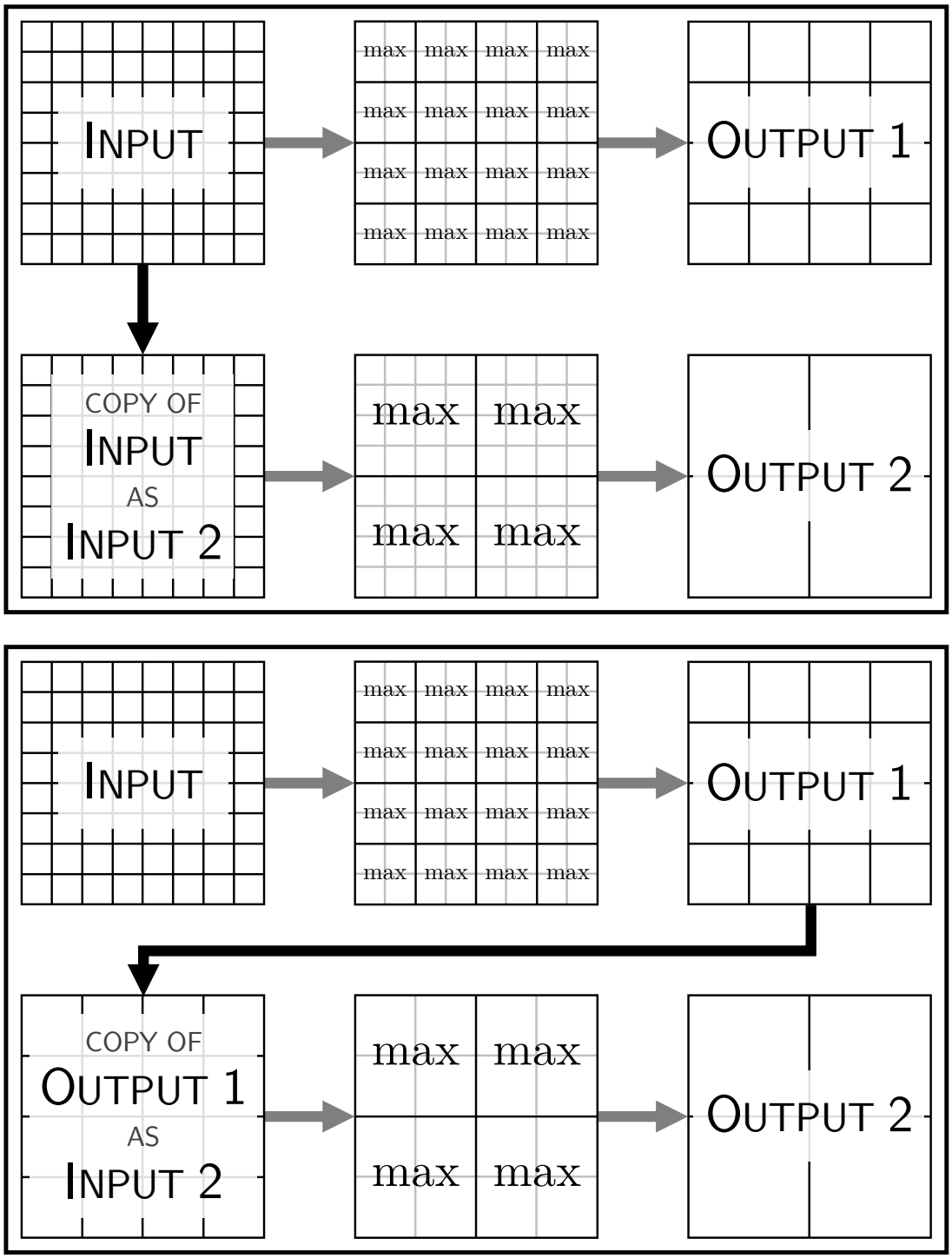


Figure 3.2: An illustrative case of image processing in W-GBC algorithm using $\max(\cdot)$ operator without (upper part) and with (lower part) recursive downsampling technique applied for an image with side length $l \doteq 2^m \doteq 2^3 = 8$ and with two box grid side lengths presented: $\varepsilon \doteq 2^{k_1} \doteq 2^1 = 2$ (odd rows) and $\varepsilon \doteq 2^{k_2} \doteq 2^2 = 4$ (even rows). Note that 'copy' may be implemented here as a *reference* (in fact the actual original object may be used) to further reduce memory overhead

alternatively, in terms of input image side length $l \doteq l(m) \doteq 2^m$:

$$n_1(l) = l^2(\log_2 l + 1). \tag{3-24b}$$

Conversely, the number $n_2(m)$ of entries in analogous situation, but with recurisve downsampling applied, is reduced in a way that the first step ($k = 0$) is the same (i.e., full input image matrix is used), but in each of the following steps matrix is downsampled that quarters the number of its entries with respect to a previous step: $L_k \doteq l_k^2 = 2^{2(m-k)}$. Thence:

$$n_2(m) = \sum_{k=0}^{m} 2^{2(m-k)} = \sum_{k=0}^{(m+1)-1} 4^m (\frac{1}{4})^k = 4^m \frac{1 - (\frac{1}{4})^{m+1}}{1 - \frac{1}{4}} = \frac{4^{m+1} - 1}{3}; \tag{3-25a}$$

or, equivalently, in terms of l:

$$n_2(l) = \frac{4l^2 - 1}{3}. \tag{3-25b}$$

From eq. (3-24) and eq. (3-25) the ratio $n_r(m)$ of the former to the latter, i.e., relative computational gain provided by employing recursive downsampling extension, is:

$$n_r(m) \doteq \frac{n_1(m)}{n_2(m)} = 4^m(m+1) \cdot \frac{3}{4^{m+1} - 1} = \frac{m+1}{4 + 4^{-m}}; \tag{3-26a}$$

and – as above – in terms of l:

$$n_r(l) = \frac{\log_2 l + 1}{4 + l^{-2}}. \tag{3-26b}$$

It may be shown that $\lim_{m \to +\infty} \frac{n_r(m)}{m} = \frac{1}{4}$ and $\lim_{m \to +\infty}(n_r(m) - \frac{1}{4}m) = \frac{1}{4}$. Therefore, the ratio function has a right-hand-side oblique asymptote $n_{ra}(m)$ given by:

$$n_{ra}(m) = \frac{m+1}{4}; \tag{3-27a}$$

or in terms of l:

$$n_{ra}(l) = \frac{\log_2 l + 1}{4}. \tag{3-27b}$$

Equations (3-26) and (3-27) demonstrate that the computational advantage of applying recursive downsampling increases with input image size. That in-

crease is linear with respect to binary exponent m of input image side length and thus is logarithmic with respect to the image side length $l \doteq 2^m$.

Figure 3.3 depicts plots of $n_1(l)$ and $n_2(l)$ in log–log scale for $m = 0, 1, \ldots, 7$. Note that only $n_2(l)$ has an (oblique) asymptote in log–log plot due to fact that $n_1(l)$ grows too fast.

An important observation regarding fig. 3.3 may be drawn by analyzing the numbers of matrix elements being processed without vs with recursive downsampling technique. Specifically, the relative profit $p_r(\cdot)$ (expressed as ratio of number of elements spared by applying the technique to number of elements processed without it applied) asymptotically – i.e., as image side length grows to infinity – – reaches $p_{ra}(\cdot) \cdot 100\% = 100\%$:

$$\begin{aligned} p_{ra}(m) &\doteq \lim_{m\to+\infty} p_r(m) \doteq \lim_{m\to+\infty} \frac{n_1(m) - n_2(m)}{n_1(m)} = \\ &= \lim_{m\to+\infty} \left(1 - \frac{n_2(m)}{n_1(m)}\right) = 1 - \lim_{m\to+\infty} \frac{n_2(m)}{n_1(m)} \doteq \\ &\doteq 1 - \lim_{m\to+\infty} \frac{1}{n_r(m)} \overset{(3\text{-}26a)}{=} 1 - \lim_{m\to+\infty} \frac{4 + 4^{-m}}{m + 1} = 1 - 0 = 1; \end{aligned} \tag{3-28a}$$

or in terms of l:

$$\begin{aligned} p_{ra}(l) &\doteq \lim_{l\to+\infty} p_r(l) \doteq \lim_{l\to+\infty} \frac{n_1(l) - n_2(l)}{n_1(l)} = \\ &= \lim_{l\to+\infty} \left(1 - \frac{n_2(l)}{n_1(l)}\right) = 1 - \lim_{l\to+\infty} \frac{n_2(l)}{n_1(l)} \doteq \\ &\doteq 1 - \lim_{l\to+\infty} \frac{1}{n_r(l)} \overset{(3\text{-}26b)}{=} 1 - \lim_{l\to+\infty} \frac{4 + l^{-2}}{\log_2 l + 1} = 1 - 0 = 1. \end{aligned} \tag{3-28b}$$

Moreover, as it is mentioned above, the primary notation is more formally complete and also more compact. Notwithstanding, their downsampling-based counterparts tend to be easily implemented. For example, due to fact that the first parts of eq. (3-13) remains identical for BBC as well as P-GBC and W-GBC variant there is no need to substitute the main box-counting routine. Hence, such implementation of FD estimator would be readily to switch between different BC sub-methods (i.e., use conditional choice of the procedure assessing the contribution of the box to N_k) as it would be possible to achieve it by only exchanging the internal downsampling methods, q.v. eqs. (3-29) to (3-32).

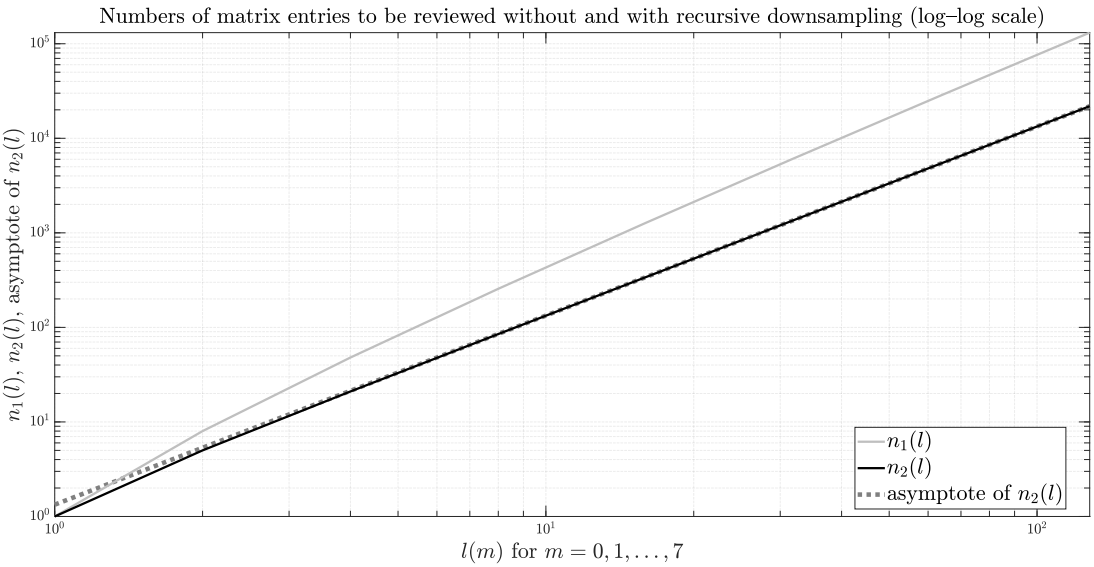


Figure 3.3: The numbers of matrix entries to be reviewed by max-based W-GBC algorithm implementations – without – $n_1(l)$ –and with –$n_2(l)$ –the recursive downsampling technique applied – as a function of the side length l of the input image $\mathbf{A}$.

Binary box-counting

For binary images, number N_k of boxes that are counted as covering the object is calculated using the following downsampling function:

$$g_{BBC}(\cdot) \doteq \text{any}(\cdot). \tag{3-29}$$

The downsampled image matrix $\mathbf{A}_k$ is obtained from $\mathbf{A}$ or $\mathbf{A}_{k-1}$ (in case of recursive downsampling for $k > 0$). This is performed in such a way that any of the pixels $x_{i,k}$ of $\mathbf{A}_k$ contains a value equal to the contribution of the corresponding box $\mathbf{B}_{i,k}$ of $\mathbf{A}$ to N_k.

In case of binary images with a zero value for the background and the value of one for the object, operator $\text{any}(\cdot)$ may be equivalently substituted by its $\max(\cdot)$ counterpart.

EDGE BINARY BOX-COUNTING This is technically a directly-derived variation of BBC that diverges from it only with respect to the pre-processing stage. For details and motivation behind its introduction, refer to section 3.2.2. Hence, the down-

sampling function of EBBC fractal-based descriptor of the images is of no difference with the one of BBC stated in eq. (3-29):

$$g_{\mathrm{EBBC}}(\cdot) \doteq g_{\mathrm{BBC}}(\cdot) \doteq \mathrm{any}(\cdot). \tag{3-30}$$

Probabilistic grayscale box-counting

For the probabilistic variant of the method utilizing grayscale image data, number N_k of boxes that are counted as covering the object is calculated in an identical procedure as in the case of BBC:

$$g_{\mathrm{P\text{-}GBC}}(\cdot) \doteq H_r(\cdot). \tag{3-31}$$

Weighted grayscale box-counting

For the weighted variant of the method utilizing grayscale images, number N_k of boxes that are counted as covering the object is calculated analogically:

$$g_{\mathrm{W\text{-}GBC}}(\cdot) \doteq \mu_p(\cdot). \tag{3-32}$$

Particularly, in the case of W-GBC algorithm with appropriate side length sizes of the image and the boxes, eqs. (3-13) and (3-32) can be efficaciously implemented by keeping the image matrix form during the calculations. This can be performed over the whole image matrix without division into individual blocks representing single boxes. Let $p \in \mathbb{R}_{\neq 0}$ and the image be of size $l \times l$, where $l = \varepsilon l_k$ ($\varepsilon, l_k \in \mathbb{N}_{>0}$). Then, $\boldsymbol{A}_k$ may be derived from $\boldsymbol{A}$ as follows:

$$\boldsymbol{A}_k = \left(\mathbf{C}_k \boldsymbol{A}^{\circ p} \mathbf{C}_k^{\mathrm{T}}\right)^{\circ \frac{1}{p}}, \tag{3-33a}$$

where $\boldsymbol{A}_k$ has dimensions $l_k \times l_k$ while $\mathbf{C}_k$ has dimensions $l_k \times \varepsilon l_k = l_k \times 2^k l_k$ whereas $\boldsymbol{A}^{\circ p}$ has dimensions $\varepsilon l_k \times \varepsilon l_k = 2^k l_k \times 2^k l_k$ and $\mathbf{C}_k^{\mathrm{T}}$ has dimensions $\varepsilon l_k \times l_k = 2^k l_k \times l_k$; also, $(\cdot)^{\circ p}$ denotes p-th Hadamard (also known as Schur or *entrywise*) power whereas:

$$\mathbf{C}_k \doteq (c_{i,j,\varepsilon})_{l_k \times \varepsilon l_k}, \tag{3-33b}$$

where:

$$c_{i,j,\varepsilon} \doteq \begin{cases} \frac{1}{\varepsilon} & \text{for } j = \varepsilon(i-1)+n \qquad (n = 1,2,\ldots,\varepsilon) \text{ or} \\ 0 & \text{otherwise} \end{cases} \tag{3-33c}$$
$$(i = 1,2,\ldots,l_k \qquad j = 1,2,\ldots,\varepsilon l_k);$$

while – because $\varepsilon \doteq 2^k$ – this notation may be simplified to:

$$\mathbf{C}_k \doteq (c_{i,j,k})_{l_k \times \varepsilon l_k}, \tag{3-33d}$$

where:

$$c_{i,j,k} \doteq \begin{cases} 2^{-k} & \text{for } j = 2^k(i-1)+n \qquad (n = 1,2,\ldots,2^k) \text{ or} \\ 0 & \text{otherwise} \end{cases} \tag{3-33e}$$
$$(i = 1,2,\ldots,l_k \qquad j = 1,2,\ldots,2^k l_k),$$

noting that both these conventions of defining $\mathbf{C}_k$ elements are equivalrnt:

$$c_{i,j,\varepsilon} \doteq c_{i,j,k} \tag{3-33f}$$

and thus this equvalence is valid also for the definition of the whole matrix $\mathbf{C}_k$:

$$\mathbf{C}_k \doteq (c_{i,j,\varepsilon})_{l_k \times \varepsilon l_k} \doteq (c_{i,j,k})_{l_k \times \varepsilon l_k}. \tag{3-33g}$$

Implementation

Listing C.1 contains the most important excerpts from MATLAB's source code implementing BC function for FDE procedure. Listing D.1 presents result of P-GBC-IMG algorithm implementation validation using illustrative input data of an image matrix having 4 px × 4 px size. MATLAB's script input code for this procedure is presented by listing C.2.

3.3 RESULTS

A set of 12 synthetic test images has been used as depicted by fig. 3.4 for experiments involving FRACTAL DIMENSION estimation using image-probability-space-based vs intensity-probability-space-based vs weighted vs differential box-counting approaches. Except for Sierpinski's triangles, all of the images, i.e., figs. 3.4a to 3.4i,

are of size 250 px × 250 px; while for Sierpinski's triangles, i.e., figs. 3.4j to 3.4l, the size is 3535 px × 3535 px.

Figure 3.4b and fig. 3.4c contain single foreground pixels directly at their left and right edges. Their purpose is auxiliary. It may be shown that adding an empty frame around an image that is subject to FDE by BC results in introducing a negative offset in the significant part of its $N_\varepsilon(\varepsilon)$ characteristic. At the same time it should not change the slope of the characteristic whose value is to be determines an the estimate of FD, but then the linear regression that is to be performed need to be unconstrained in terms of intersecting the origin of the coordinate system. Nevertheless, when plotting characteristics of a few images on a single graph, it is not desired to have large vertical shifts between them, as it hinders the possibility of visually evaluating and comparing them. Therefore, a mechanism of removing the empty borders and rescaling the image to square proportions has been implemented in some implementations of the FDE procedure to possibly reduce the offsets and facilitate the visual validation of the results. This works well for most cases of the typical images except for these having extremely small size in at least one of the dimensions. E. g., a signle pixel would be extracted by this mechanism and rescaled to form a full square, and thus its dimension estimate could grow from a value slightly above zero to two. The same would happen for a section with an orientation close to horizontal or vertical. As a simple way to prevent this disruption of the results, the addition of the image-stretching pixels has been chosen. This prevents the abovementioned problem when at the same time has negligible influence on the estimated dimension value. Namely, in the considered range of scales, objects being significantly larger than single pixel would have a positive dimension that would dominate the result (as an union of a figure with two single points has the same dimension as the figure). Alternatively, the object itself would be of near-zero dimension – and such tiny or almost completely empty objects are effectively rendered as very small collections of pixels – so it would belong to the same dimensional class as the image-stretching pixels (as an union of a few single points with two additional points has still a dimension of a single point). In either case, such stretching shall not have any significant impact on the result that has been verified experimentally. Not always has the implementation with the described frame reduction been employed during the research, nevertheless, to maintain consistency, the two images resembling the vertical sections (in fig. 3.4b having width of 1 px and in fig. 3.4c – of 3 px that thins to 1 px in the bottom end) always have had the stretching pixels added.

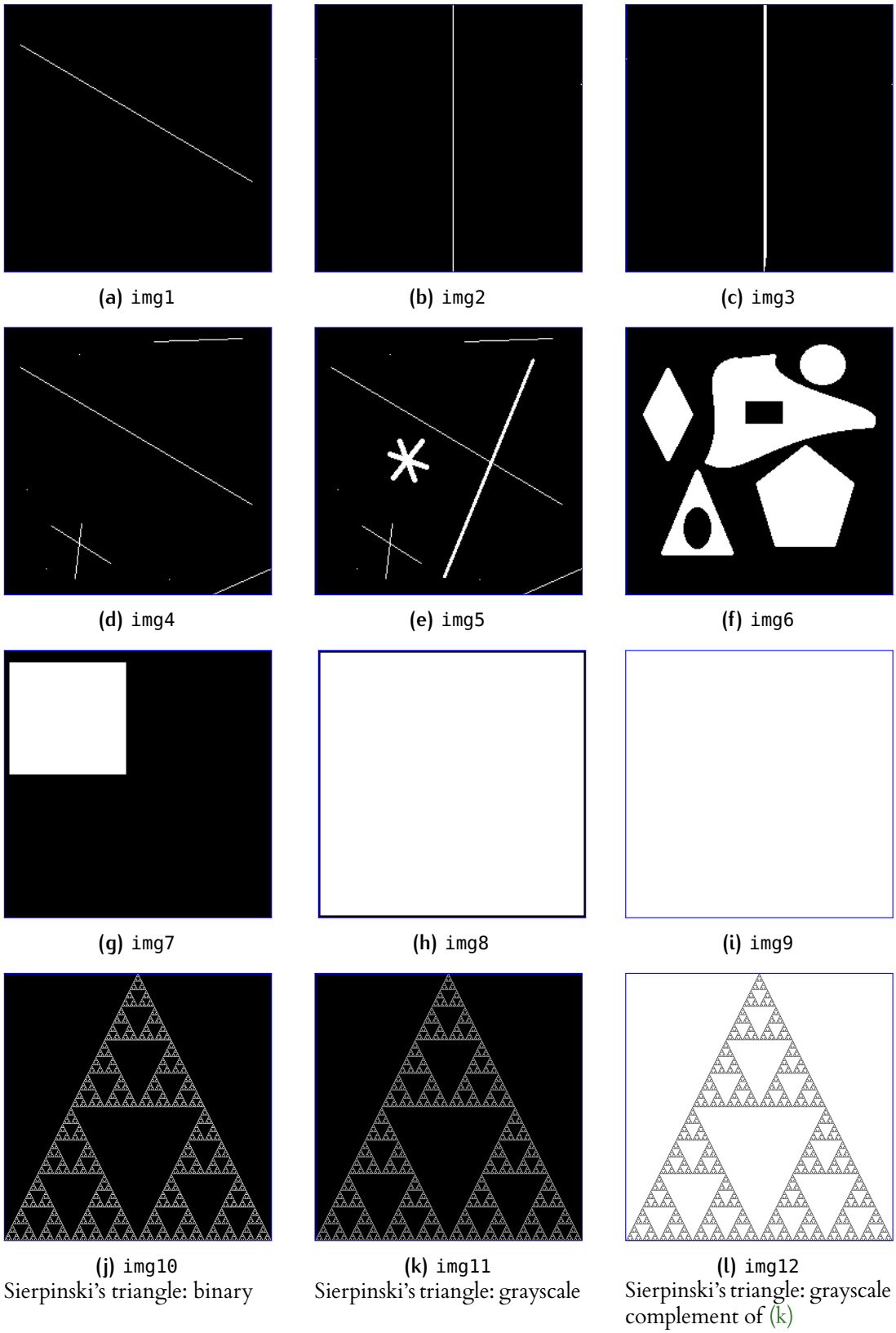
Sierpinski's triangle has been visualised in three versions of which two approximations, i.e., binary, q.v. fig. 3.4i, as well as grayscale – generated by downscaling a binary image of a much larger resolution to obtain the result shown in fig. 3.4j – are expected to have their FDs estimated by a value close to the self-similarity dimension – or to Hausdorff's dimension, that in this case has identical value – of the original theoretical Sierpinski's set. This reference value is equal to $\log(3)/\log(2) = \log_2(3) \approx 1.5850$. On the contrary, complementary sets for both cases, of which the grayscale complement, q.v. fig. 3.4k), has been utilized during research, are expected to have FDs of two. It is so due to the fact that the complement of Sierpinski's set contains a dense background composed of solid triangles that have the topological dimension of two and thus dominate the box counting result of such image. Note that for such complement two largest triangles of the foreground triangular components together occupy half of the whole image surface. Therefore, as surface is a 2-D measure and topological dimension of a polygon is equal to its FD, it is clearly visible that over 50% of the image data of fig. 3.4k represents a structure having FD of two.

Following experiments on the simulated interference data, the real data consisting of 686 gray-level interferometric images – originally of size 720 px × 576 px from which square central parts of size 512 px × 512 px have been extracted as inputs – of human tear film acquired in an *in-vivo* manner have been added to the test set. Included have been 479 interferograms of healthy subjects along with 207 affected with dry eye syndrome (DES). Three illustrative cases are depicted in fig. 3.6. Refer also to figs. 7.2 and 7.3 in section 7.1.1 that extend this illustration to facilitate extended description of LSI imagery setup and methodological grounds of analysis scheme for the acquired interferometric data.

High statistically significant correlation has been found between BBC and W-GBC: Pearson's $\rho = 0.998$, $p < 0.001$. Similarly high correlation coefficient has been obtained between BBC and PROBABILISTIC GRAYSCALE BOX-COUNTING based on image probability space (P-GBC-IMG): $\rho = 0.993$, $p < 0.001$. It has not been the case between BBC and PROBABILISTIC GRAYSCALE BOX-COUNTING based on intensity probability space (P-GBC-INT): $\rho = 0.365$, $p < 0.001$ – cf. fig. 3.5 and fig. 3.7 where black, blue and red marks denote data from synthetic, normal and DES subjects, respectively. Also, a good agreement between BBC and other estimators, for both synthetic and real images, has been achieved only for W-GBC.

The discriminative powers of FD estimators have been also evaluated. For this purpose, the results of each considered FD estimator for the images from healthy sub-

jects have been contrasted against their corresponding FD estimators for the images from DES group. Receiver operating characteristic (ROC) curves have been evaluated using kernel density estimators and areas under the curves (AUCs) have been estimated. Figure 3.8 shows the results for BBC – AUC of 63.8% – and W-GBC – – AUC of 69.7% – the two FD estimators that achieved the highest discriminating power. For P-GBC-IMG, P-GBC-INT and DBC the discriminating powers have not been satisfactory, achieving AUCs of 51.3%, 53.4% and 51.2%, respectively.



(a) `img1` (b) `img2` (c) `img3`

(d) `img4` (e) `img5` (f) `img6`

(g) `img7` (h) `img8` (i) `img9`

(j) `img10` Sierpinski's triangle: binary

(k) `img11` Sierpinski's triangle: grayscale

(l) `img12` Sierpinski's triangle: grayscale complement of (k)

Figure 3.4: The set of 12 synthetic images used synthetic images used for testing fractal dimension estimation algorithms. Blue frames imposed around each subfigure's borders illustrate images' edges and are not part of actual image data

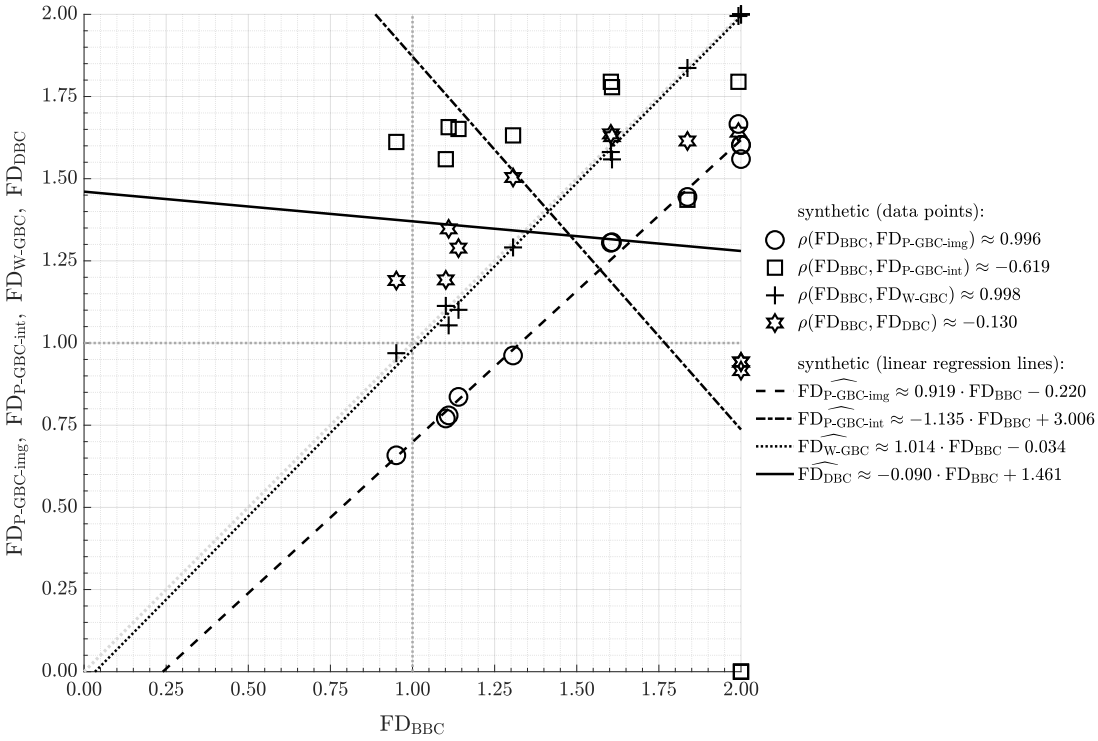
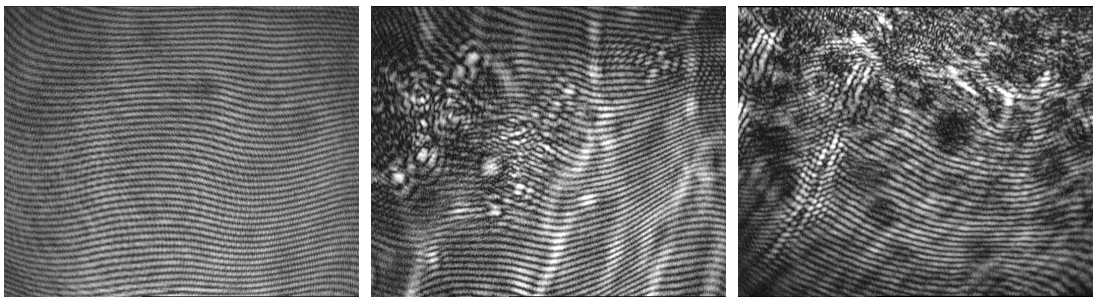
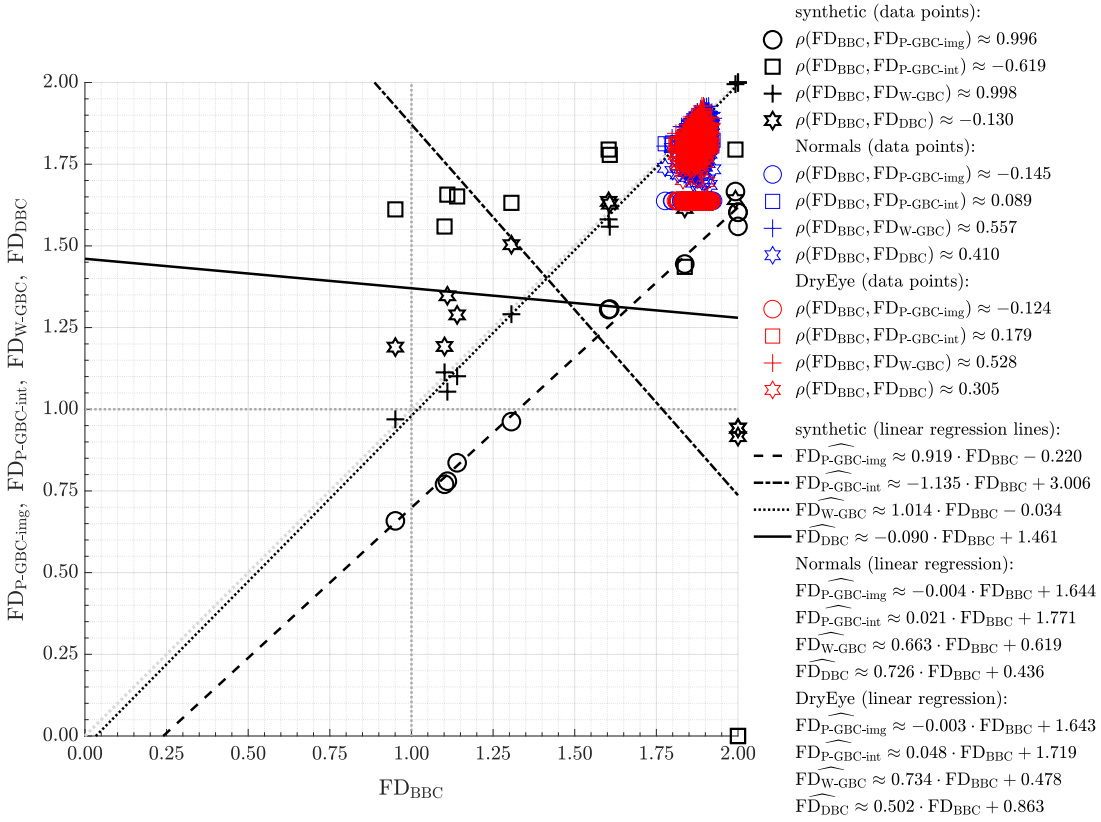


Figure 3.5: Comparison of FD estimates of 12 synthetic images presented in fig. 3.4 calculated using different methods: $\mathrm{FD}_{\mathrm{P\text{-}GBC\text{-}IMG}}$, $\mathrm{FD}_{\mathrm{P\text{-}GBC\text{-}INT}}$, $\mathrm{FD}_{\mathrm{W\text{-}GBC}}$ and $\mathrm{FD}_{\mathrm{DBC}}$ with $\mathrm{FD}_{\mathrm{BBC}}$

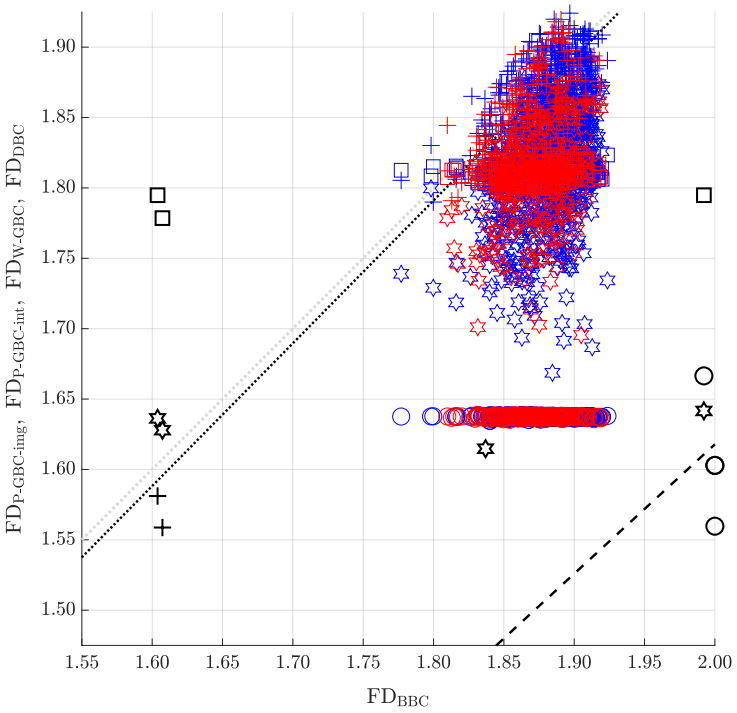


(a) Good quality **(b)** Moderate quality **(c)** Poor quality

Figure 3.6: Three illustrative real images of human tear film acquired in an *in vivo* fashion

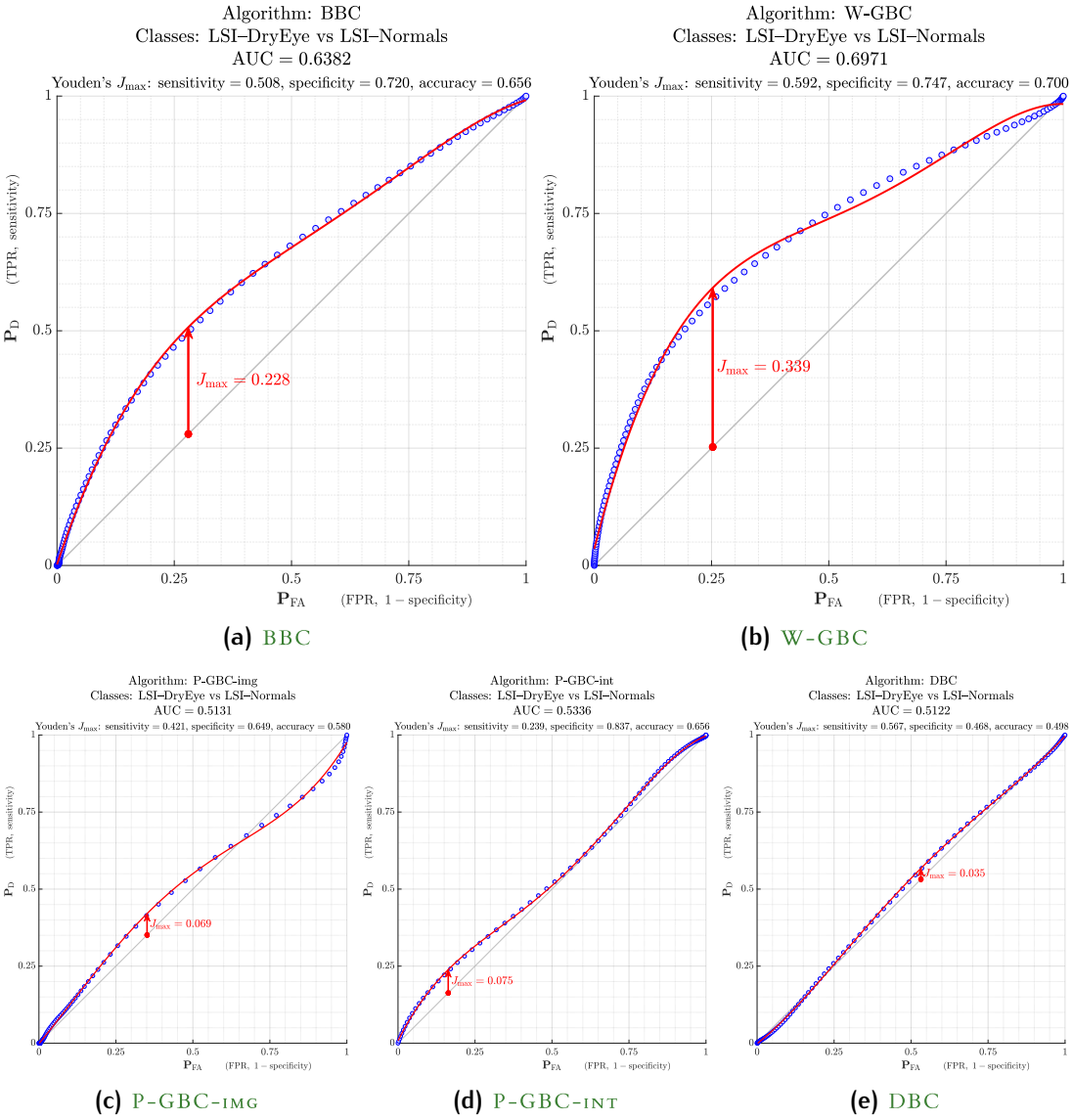


(a) Synthetic and real images



(b) Synthetic and real images – enlargement of the clusters of real data FD estimates

Figure 3.7: Comparison of FD estimates from fig. 3.5 with superimposed results for real images (blue and red marks for healthy and DES-affected subjects, respectively)



(a) BBC **(b)** W-GBC

(c) P-GBC-IMG **(d)** P-GBC-INT **(e)** DBC

Figure 3.8: ROC curves for different BC-based algorithms – discriminative powers for distinguishing DES-affected from healthy individuals using LSI images

3.4 PARTIAL CONCLUSIONS

The mathematical notation used in primary definitions is more compact in a sense that it does not involve the downsampling process that would substantially complicate the formal notation. However, by using proper downsampling implementation exploiting the recursive idempotence property, when possible, a substantial drop in computational complexity might be achieved.Thus, the recursive downsampling technique and theorem 1 relating to it may be indicated as an important innovation to the methodology of BC calculation. Additionally, a simple example of downsampling function g having the property of recursive idempotence (and thus applicable to theorem 1) is presented in example 1 and an instance of a practical benefit achieved by applying recursive downsampling technique to image processing is illustrated in fig. 3.2.

Points of coordinates $(2.00, 0.00)$ depicted in figs. 3.5 and 3.7a occurs for sets representing monochromatic images of squares, i.e., image data from figs. 3.4g to 3.4i (the same applies to fig. 7.16a). This is caused by inconsistency of P--GBC-INT with other BC-based algorithms. Specifically, for image-probability--space-based algorithms a solid square of any non-background value is treated as an object of FD that is estimated to a value close to its Euclidean dimension, i.e., 2. On the contrary, for intensity-probability-space-based P-GBC-INT, such object is represented as a uniform surface, i.e., it is flat, and such lack of variability in the reference space determines its FD estimate to be 0.

The study shows that entropy based on intensity fails to adequately represent FD for the set of real interference images. The proposed W-GBC FD estimator achieves superior performance to that of BBC estimator with respect to differentiating interferometry images of tear film of normal subjects from DES-affected ones. Also, W-GBC FD estimator overcomes the requirement for setting a particular threshold level during the pre-processing step. These developments indicate that the choice of a FD estimator may need to be tailored to a particular application, peculiarly when computational efficiency of an estimator is of interest or the difficulty in setting the threshold level for BC-based FD estimator is encountered. Examples of such situations include image analysis cases when the variability of the background intensity is high or the contrast level between the foreground and the background is low.

It is worth noting that off-the-shelf techniques, each having its advantages and limitations, for estimating FD of an image may not always provide satisfac-

tory results when FD-based classification is considered. Hence, each application involving certain class of images need to be carefully considered.

4 GLOBAL COUNTERPARTS OF P-GBC-IMG AND P-GBC-INT

4.1 MOTIVATION

There are several other approaches to fractal dimension estimation (FDE) that consider *global* measures. In this short chapter the literature review gap is filled to cover the global variant of probabilistic grayscale box-counting (P-GBC) approach. The *global* approach to P-GBC is utilized in the concept of *generalized dimensions* to address an issue that a single number, i.e., fractal dimension (FD) estimate, is sometimes not a complete descriptor of a set. E.g., this is the case in modeling of dynamical systems and other usually at least incompletely self-similar objects [76, 77]. This approach might be formulated in terms of Hausdorff's or box-counting (BC) dimensions, from which the latter is also called *capacity dimension* and is more of a practical use [76] and complement with the notation used in this thesis. There are infinitely many generalized FDs denoted D_q and they are differentiated by their numbers – with q being assigned to q-*th generalized dimension* – as well as – for chosen cases – also by their names [78]. E.g., first generalized dimension is called *information dimension* [76] and its definition include mathematical operation identical to calculating of information (Shannon's) entropy [77]. As P-GBC in both probability spaces, i.e., spatial and intensity ones, is also based on information entropy, it is worth noting that its global variants P-GBC-IMG-GLB and P-GBC-INT-GLB are instances of the aforementioned information dimension. It is referred to in literature [76, 77] what makes it common enough to be of importance justifying its mentioning in this thesis, despite the fact that the work is based mostly on the *local approach* to P-GBC which is more loosely related to generalized dimensions. Also, their concepcion – and thus the global approach to P-GBC – is also related to the concept of *multifractality* [77] used in *multifractal analysis* that constitutes a broader term than *fractal analysis* regarded in this thesis.

4.2 CALCULATING P-GBC-IMG-GLB AND P-GBC-INT--GLB

Global variant of P-GBC approach, further referred to in an abbreviated manner by both of its variations, namely P-GBC-IMG-GLB and P-GBC-INT-GLB, is described using *primary notation* as:

$$N_{k,\,\mathrm{glb.}}(\mathbf{A}) \doteq H_{r,\,\mathrm{glb.}}(\mathbf{A}, k) \doteq \sum_{i=1}^{c_k} p_{\cdot}(\mathbf{B}_{i,k}) \log_r \frac{1}{p_{\cdot}(\mathbf{B}_{i,k})} \qquad (r \in \mathbb{R}_{>0}), \quad (4\text{-}1a)$$

i.e., *box-wise* image entropy calculated for k-th box division. The above expression may be expressed also by using *global downsampling-based* convention:

$$N_{k,\,\mathrm{glb.}}(\mathbf{A}_k) \doteq h_{r,\,\mathrm{glb.}}(\mathbf{A}_k) \doteq \sum_{i=1}^{c_k} x_{i,k} \log_r \frac{1}{x_{i,k}} \qquad (r \in \mathbb{R}_{>0}), \quad (4\text{-}1b)$$

where – by analogy to eq. (3-13):

$$x_{i,k} \doteq g(\mathbf{B}_{i,k}) \quad (4\text{-}1c)$$

and – by combining eqs. (4-1a) and (4-1c):

$$g(\mathbf{B}_{i,k}) \doteq p_{\cdot}(\mathbf{B}_{i,k}). \quad (4\text{-}1d)$$

In this formulation, $h_{r,\,\mathrm{glb.}}(\mathbf{A}_k) \doteq H_{r,\,\mathrm{glb.}}(\mathbf{A}, k)$ denotes image *global* entropy calculated using k-th downsampled image employing base-r logarithm. Probability space of these functions may be of two kinds. It may be defined as *spatial*: $H_{r,\,\mathrm{glb.}}(\cdot, \cdot) \doteq H_{r,\,\mathrm{glb.,\,img.}}(\cdot, \cdot) \in [0, 2\log_r \varepsilon]$, whereas it may be also *intensity*: $H_{r,\,\mathrm{glb.}}(\cdot, \cdot) \doteq H_{r,\,\mathrm{glb.,\,int.}}(\cdot, \cdot) \in [0, \log_r U]$ – depending on the choice of the internal probability calculation sub-variant: spatial-probability-space- or intensity--probability-space-based, respectively. $H_{r,\,\mathrm{glb.,\,img.}}(\cdot, \cdot)$ and $H_{r,\,\mathrm{glb.,\,int.}}(\cdot, \cdot)$ utilize *cumulative probabilities of boxes* $p_{\mathrm{img.}}(\mathbf{B}_{i,k})$ and probability $p_{\mathrm{int.}}(\mathbf{B}_{i,k})$, respectively.

4.3 SPATIAL-PROBABILITY-SPACE-BASED GLOBAL ENTROPY

In case of global P-GBC method variant based on probability defined in *spatial (image)* probability space, its name is abbreviated as P-GBC-IMG-GLB. Then, cumulative

(i.e., box-range-summarized) and pixel-range *global* (i.e., calculated with respect to the matrix of the whole image) photon–pixel incidence probabilities definitions are described by eq. (4-2):

$$p_{\text{img.}}(\mathbf{B}_{i,k}) \doteq \sum_{j=1}^{\varepsilon^2} p_{\text{img., glb.}}(x_{i,j,k}) \in [0, \varepsilon^2], \tag{4-2a}$$

$$p_{\text{img., glb.}}(x_{i,j,k}) \doteq \frac{x_{i,j,k}}{\sum_{s=1}^{c_k} \sum_{t=1}^{\varepsilon^2} x_{s,t,k}} \in [0, 1]. \tag{4-2b}$$

4.4 INTENSITY-PROBABILITY-SPACE-BASED GLOBAL ENTROPY

In case of probability calculated in *intensity* probability space, a variant of global P-GBC abbreviated as P-GBC-INT-GLB, cumulative probabilities of boxes and – – what is important – *global* ***box-context*** pixel-range histogram bin probabilities are stated by eq. (4-3):

$$p_{\text{int.}}(\mathbf{B}_{i,k}) \doteq \sum_{u=1}^{U} p_{\text{int. glb.}}(b_u, i, k) \in [0, S], \tag{4-3a}$$

$$p_{\text{int. glb.}}(b_u, i, k) \doteq \frac{\sum_{s=1}^{c_k} \sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x : x \in b_u \wedge \exists y \text{ in } \mathbf{B}_{i,k}.y \in b_u\}}(x_{s,t,k})}{c_k \varepsilon^2} \in [0, 1] \tag{4-3b}$$

or – equivalently:

$$p_{\text{int. glb.}}(b_u, i, k) \doteq \frac{\sum_{s=1}^{c_k} \sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x : x \in b_u \cap b_{i,k}\})}(x_{s,t,k})}{c_k \varepsilon^2} \in [0, 1], \tag{4-3c}$$

where $b_{i,k}$ constitutes a union of all histogram bins that are represented in box $\mathbf{B}_{i,k}$:

$$b_{i,k} \doteq \bigcup \{ b_u : u \in \overline{1, U} \wedge \exists x \text{ in } \mathbf{B}_{i,k}. x \in b_u \}. \tag{4-3d}$$

Here, the probability $p_{\text{int.glb.}}(b_u, \cdot, \cdot)$ of histogram bin b_u is *global* (image-wise) but expressed in *box context*. This means that it is by definition fixed to be zero for any histogram bin that has no representatives within the considered box $\mathbf{B}_{i,k}$. Similarly as for *local* variant of probability, this value is zero even in cases where the bin in question does possess pixel exemplification(s) in other boxes of the image (and thus has non-zero probability in context of these boxes). Contrary to *local* probability, though, for all bins represented within the box at least once, this variant has global (image-wise) scope. Therefrom, in such cases its value is equal to a *typical* case of *global* (image-wise) probability. This typical probability, in turn, in a more explanatory (while slightly redundant) manner with respect to such *contextual* distinction would be termed *global* ***image-context***. Concluding, *global* ***box-context*** probability from eq. (4-3c) might be characterized as being equal to *local probability* for bins that do not occur in given box while otherwise being equal to *global probability*. The former – in this usage – is always zero as probability of an impossible event is zero by definition and in a given box occurence of a pixel from bin that has no pixels in that box is – also by definition – an impossible event. Therefore, full definition of *local probability*– althought simple – need not to be recounted here, whereas it is defined above in eq. (3-10b). The latter of mentioned probabilities, namely the **purely** *global* one, is, contradistically, defined without any reference to box number i. This definition is presented in eq. (4-3e) above.

Purely *global*, i.e., specified in the **image-context**, probability of the histogram bin b_u is defined as:

$$p_{\text{int.,glb.,pure}}(b_u) \doteq \frac{\sum_{s=1}^{c_k} \sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x : x \in b_u\}}(x_{s,t,k})}{c_k \varepsilon^2} \in [0, 1]. \tag{4-3e}$$

4.4.1 Remarks on contexts of global probability in the intensity probability space

Note that:

$$0 \leqslant \mathbb{1}_{\{x : x \in b_u \wedge \exists y \text{ in } \mathbf{B}_{i,k}.\, y \in b_u\}}(\cdot) \leqslant \mathbb{1}_{\{x : x \in b_u\}}(\cdot) \leqslant 1. \tag{4-4a}$$

Due to eq. (4-4a), eq. (3-10b) means that probability of u-th bin in *box context* always, i.e., $\forall u \in \overline{1, U}. \forall k \in \overline{0, m}. \forall i \in \overline{0, c_k}$, equals at most its probability in *purely global* context:

$$0 \leqslant p_{\text{int.,glb.}}(b_u, i, k) \leqslant p_{\text{int.,glb.,pure}}(b_u) \leqslant 1. \tag{4-4b}$$

Specifically, *box-context global* probability $p_{\text{int.,glb.}}(b_u, i, k)$ of u-th histogram bin b_u equals *purely global* probability $p_{\text{int.,glb.,pure}}$ if and only if there is a pixel y within box $\mathbf{B}_{i,k}$ that falls into the scope of this histogram bin; otherwise it is equal to zero:

$$p_{\text{int.,glb.}}(b_u, i, k) = \begin{cases} p_{\text{int.,glb.,pure}}(b_u) & \exists y \text{ in } \mathbf{B}_{i,k}. y \in b_u \text{ or} \\ 0 & \text{otherwise} \end{cases}. \tag{4-4c}$$

4.4.2 Remarks on the unusability of purely global probability in the intensity probability space

Note that probability $p_{\text{int.,glb.}}(b_u, i, k)$ is not *purely global*, as calculation of cumulative box probability as such, i.e., without *box context*:

$$p_{\text{int.,pure}}(\mathbf{B}_{i,k}) \doteq \sum_{u=1}^{U} p_{\text{int.,glb.,pure}}(b_u), \tag{4-5a}$$

would always yield the result equal to one as demonstrated by theorem 2.

Theorem 2. The unusability of purely global probability in the intensity probability space:
The purely global probability in the intensity probability space is practically unusable by being identically equal to one:

$$p_{\text{int.,pure}}(\mathbf{B}_{i,k}) \equiv 1. \tag{4-5b}$$

Proof:
Reminding the definition of *purely global probability* $p_{\text{int.}}(b_u)$ from eq. (4-3e) renders:

$$p_{\text{int.,glb.,pure}}(b_u) \doteq \frac{\sum_{s=1}^{c_k} \sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x : x \in b_u\}}(x_{s,t,k})}{c_k \varepsilon^2}. \tag{4-5c}$$

This leads to:

$$\begin{aligned} p_{\text{int.,pure}}(\mathbf{B}_{i,k}) &\doteq \sum_{u=1}^{U} \frac{\sum_{s=1}^{c_k}\sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x:x\in b_u\}}(x_{s,t,k})}{c_k\varepsilon^2} = \\ &= \frac{\sum_{s=1}^{c_k}\sum_{t=1}^{\varepsilon^2}\sum_{u=1}^{U} \mathbb{1}_{\{x:x\in b_u\}}(x_{s,t,k})}{c_k\varepsilon^2} = \\ &= \frac{1}{c_k\varepsilon^2}\sum_{s=1}^{c_k}\sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x:x\in\bigcup_{u=1}^{U} b_u\}}(x_{s,t,k}) \equiv \\ &\equiv \frac{1}{c_k\varepsilon^2}\sum_{s=1}^{c_k}\sum_{t=1}^{\varepsilon^2} 1 = \frac{c_k\varepsilon^2}{c_k\varepsilon^2} \equiv 1. \end{aligned} \tag{4-5d}$$

This means that the purely global probability in the intensity probability space is identically equal to one and thus practically unusable. ■

4.4.3 Remarks on nature of box-context global probability in the intensity probability space

It is important to note the fact that ***box-context*** *global* probability is just a technical concept. It has been coined to assure the usability of P-GBC-INT-GLB algorithm as it would be rendered unusable if using ***purely*** *global* probability as it is demonstrated in eq. (4-5d). Moreover, it is named '*probability*' due to the obvious connection of its concept to other probabilities. Nonetheless, in is not a usually accepted *probability* measure per se, as it violates the second Kolmogorov's axiom stating that probability is normalized to one (unitarity) [79, 80]. Namely, one elementary event from sample space must occur. Therefore, probability of any of them occurring is by axiom equal to one. This is not the case for box-context global probability in the intensity probability space as demonstrated by lemma β.

Lemma β. Violation of the second Kolmogorov's axiom by box-context global quasi-probability:
A function termed under the conventionalized (for the sake of simplicity) name of "*box-context global probability*" violates the second Kolmogorov's axiom.

PROOF:
Suppose that in k-th ($k = 0$) box division of the image there are $c_0 = 4$ boxes, with each of them containing $\varepsilon^2 \doteq (2^k)^2 = 1$, i.e., single, pixel. Furthermore, suppose that there are $U = 4$ histogram bins in total, of which the first bin ($u = 1$)

is represented by the first three ($i = 1, 2, 3$) of four pixels in total, and the second bin ($u = 2$) is represented in the last box ($i = 4$). Then, the probability of u-th ($u = 1$) bin occurring in context of this box is:

$$\begin{aligned} p_{\text{int. glb.}}(b_1, 1, 0) &\doteq \frac{\sum_{s=1}^{4}\sum_{t=1}^{1} \mathbb{1}_{\{x : x \in b_1 \wedge \exists y \text{ in } \mathbf{B}_{1,0}.\, y \in b_1\}}(x_{s,t,0})}{4 \cdot 1} = \\ &= \frac{\sum_{s=1}^{4} \mathbb{1}_{\{x : x \in b_1 \wedge \exists y \text{ in } x_{1,1,0}.\, y \in b_1\}}(x_{s,1,0})}{4} = \\ &= \frac{\sum_{s=1}^{4} \mathbb{1}_{\{x : x \in b_1 \wedge x_{1,1,0} \in b_1\}}(x_{1,1,0})}{4} = \\ &= \frac{1+1+1+0}{4} = \frac{3}{4}; \end{aligned} \tag{4-6a}$$

because for $u = 2, 3, 4$:

$$\nexists y \text{ in } \mathbf{B}_{1,0}.\, y \in b_u \iff x_{1,1,0} \notin b_u, \tag{4-6b}$$

for other bins the probability is zero:

$$p_{\text{int. glb.}}(b_2, 1, 0) = p_{\text{int. glb.}}(b_3, 1, 0) = p_{\text{int. glb.}}(b_4, 1, 0) = 0. \tag{4-6c}$$

Therefore for event that happen by axiom surely:

$$\begin{aligned} 1 \overset{\text{def}}{=} p_{\text{int. glb.}}(\Omega, 1, 0) &\doteq p_{\text{int. glb.}}\Big(\bigcup_{u=1}^{4} b_u, 1, 0\Big) \doteq \\ &\doteq \sum_{u=1}^{4} p_{\text{int. glb.}}(b_u, 1, 0) \doteq \frac{3}{4} \neq 1, \end{aligned} \tag{4-6d}$$

which forms a contradiction. ■

Corollary 1 shows that lemma β implies that quasi-probability $p_{\text{int. glb.}}(\cdot, \cdot, \cdot)$ is not Kolmogorov's probability.

Corollary 1. Violation of the second Kolmogorov's axiom by box-context global quasi-probability measure:
The contradicition from the proof of lemma β means that the function termed ***box-context*** *global probability* is in fact a quasi-probability, or a probability-like function, but not a *probability in Kolmogorov's sense*. □

Nevertheless, theorem 3 demonstrates that the issue exhibited in corollary 1 may be mitigated by exploiting the normalization procedure that enables the unitarity property to be provided to $p_{\text{int. glb.}}(\cdot, \cdot, \cdot)$ quasi-probability.

Theorem 3. Possibility of box-context global quasi-probability normalization:
Shall it be necessary to regain unitarity by box-context global quasi-probability, i.e., to have value of the certain event fixed at one, it is possible to overcome the limitation from corollary 1 by *normalization*.

Proof:
Global quasi-probability function can be modified to regain unitarity by normalization performed the following manner:

$$\widetilde{p_{\text{int. glb.}}}(b_u, i, k) \doteq \frac{p_{\text{int. glb.}}(b_u, i, k)}{\sum_{u=1}^{U} p_{\text{int. glb.}}(b_u, i, k)}. \tag{4-6e}$$

This assures that $\widetilde{p_{\text{int. glb.}}}(b_u, i, k)(\cdot)$ can technically be termed a probability in Kolmogorov's sense, as it satisfies all the mathematical necessities for being such a measure. ■

Equation (4-6e) demonstrates that box-context global quasi-probability defined in the intensity probability space may be in general– and in a relatively simple manner – converted to Kolmogorov's probability. Thus, in effort to maintain the clarity of the text while the same terms are often recalled repeatedly, this thesis often involves references to $p_{\text{int. glb.}}(\cdot, \cdot, \cdot)$ as to 'probability' rather than 'quasi-probability' or a similar technically precise term. This simplification may introduce a minor naming imprecision, therefore it is harnessed only through the justification provided by theorem 3 together with the explanation provided by this paragraph.

4.5 Remarks on the Unusability of Local Probabilities for Global Variants of Probabilistic Grayscale Box-Counting

4.5.1 Unusability of $p_{\text{img., loc.}}(\cdot)$

It should be noted that probability $p_{\text{img.}}(\cdot)$ would be named in a fully descriptive manner as $p_{\text{img., glb.}}(\cdot)$, nevertheless the shorter notation is chosen as this thesis does not utilize possible local counterpart of this cumulative probability which under convetion used throughout this thesis would be defined by the term $p_{\text{img., loc.}}(\cdot)$. Moreover, such probability would be trivial and unusable as shown by theorem 4.

Theorem 4. The unusability of $p_{\text{img.},\,\text{loc.}}(\cdot)$:
The probability $p_{\text{img.},\,\text{loc.}}(\cdot)$ is practically unusable by being identically equal to one:

$$p_{\text{img.},\,\text{loc.}}(\mathbf{B}_{i,k}) \equiv 1. \tag{4-7a}$$

PROOF:
By definition:

$$p_{\text{img.},\,\text{loc.}}(\mathbf{B}_{i,k}) \doteq \sum_{j=1}^{\varepsilon^2} p_{\text{img.}}(x_{i,j,k}). \tag{4-7b}$$

Further, basing on eq. (3-9b) defining $p_{\text{img.}}(x_{i,j,k})$:

$$p_{\text{img.},\,\text{loc.}}(\mathbf{B}_{i,k}) \doteq \sum_{j=1}^{\varepsilon^2} \frac{x_{i,j,k}}{\sum_{t=1}^{\varepsilon^2} x_{i,t,k}} = \frac{\sum_{j=1}^{\varepsilon^2} x_{i,j,k}}{\sum_{t=1}^{\varepsilon^2} x_{i,t,k}} \equiv 1. \tag{4-7c}$$

This means that the probability $p_{\text{img.},\,\text{loc.}}(\cdot)$ is identically equal to one and thus practically unusable. ∎

4.5.2 Usability of $p_{\text{img.}}(\cdot)$

This is not the case for $p_{\text{img.}}(\cdot) \doteq p_{\text{img.},\,\text{glb.}}(\cdot)$, because not necessarily all photons coincide with pixels of a single box. That is, global probabilities of pixels within a single box not necessarily sum up to one and their sum might be smaller as shown by theorem 5.

Theorem 5. The usability of $p_{\text{img.}}(\cdot) \doteq p_{\text{img.},\,\text{glb.}}(\cdot)$:
The probability $p_{\text{img.}}(\cdot) \doteq p_{\text{img.},\,\text{glb.}}(\cdot)$ is practically usable in the sense of not being identically equal to one:

$$p_{\text{img.}}(\mathbf{B}_{i,k}) \doteq p_{\text{img.},\,\text{glb.}}(\mathbf{B}_{i,k}) \not\equiv 1. \tag{4-8a}$$

PROOF:
For k-th ($k \in \overline{0, m}$) box division and i-th ($i \in \overline{0, c_k}$) box $\cdot$, basing on eq. (4-2b) and the possibility of existence of a case (or cases) of $x_{s,t,k} > 0$ ($s \neq i$), the following is true:

$$1 \not\equiv \frac{\sum_{j=1}^{\varepsilon^2} x_{i,j,k}}{\sum_{s=1}^{c_k} \sum_{t=1}^{\varepsilon^2} x_{s,t,k}} \leqslant \frac{\sum_{j=1}^{\varepsilon^2} x_{i,j,k}}{\sum_{t=1}^{\varepsilon^2} x_{i,t,k}} \equiv 1. \tag{4-8b}$$

Therefore, eq. (4-7c) from the proof of theorem 4 fails for *global* probability in the image probability space $p_{\text{img.}, \text{glb.}}(\cdot)$:

$$p_{\text{img.}}(\mathbf{B}_{i,k}) \doteq p_{\text{img.}, \text{glb.}}(\mathbf{B}_{i,k}) \not\equiv 1. \tag{4-8c}$$

The above proves the usability of $p_{\text{img.}}(\cdot)$. ■

4.5.3 Unusability of $p_{\text{int.}, \text{loc.}}(\cdot)$

Similarly, $p_{\text{int.}}(\cdot)$ would be named in a more specific manner as $p_{\text{int.}, \text{glb.}}(\cdot)$, whereas no instance of $p_{\text{int.}, \text{loc.}}(\cdot)$ is used in this thesis. Therefore, the shortened notation is chosen again. Furthermore, this probability $p_{\text{int.}, \text{loc.}}(\cdot)$ would be trivial and unusable as shown by theorem 6.

Theorem 6. The unusability of $p_{\text{int.}, \text{loc.}}(\cdot)$:
The probability $p_{\text{int.}, \text{loc.}}(\cdot)$ is practically unusable by being identically equal to one:

$$p_{\text{int.}, \text{loc.}}(\mathbf{B}_{i,k}) \equiv 1. \tag{4-9a}$$

Proof:
By definition:

$$\begin{aligned} p_{\text{int.}, \text{loc.}}(\mathbf{B}_{i,k}) &\doteq \sum_{u=1}^{U} p_{\text{int.}}(b_u, i, k) \doteq \\ &\doteq \sum_{u=1}^{U} \frac{\sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x : x \in b_u\}}(x_{i,t,k})}{\varepsilon^2} = \\ &= \frac{\sum_{t=1}^{\varepsilon^2} \sum_{u=1}^{U} \mathbb{1}_{\{x : x \in b_u\}}(x_{i,t,k})}{\varepsilon^2} = \\ &= \frac{\sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x : x \in \bigcup_{u=1}^{U} b_u\}}(x_{i,t,k})}{\varepsilon^2} \equiv \frac{\sum_{t=1}^{\varepsilon^2} 1}{\varepsilon^2} \equiv 1. \end{aligned} \tag{4-9b}$$

This means that the probability $p_{\text{int.}, \text{loc.}}(\cdot)$ is identically equal to one and thus practically unusable. ■

4.5.4 Usability of $p_{\text{int.}}(\cdot)$

This is not the case for $p_{\text{int.}}(\cdot) \doteq p_{\text{int.}, \text{glb.}}(\cdot)$, because not necessarily all bins of the histogram are represented in each box as shown by theorem 7.

Theorem 7. The usability of $\mathrm{p}_{\text{int.}}(\,\cdot\,) \doteq \mathrm{p}_{\text{int., glb.}}(\,\cdot\,)$:
The probability $\mathrm{p}_{\text{int.}}(\,\cdot\,) \doteq \mathrm{p}_{\text{int., glb.}}(\,\cdot\,)$ is practically usable in the sense of not being identically equal to one:

$$\mathrm{p}_{\text{int.}}(\mathbf{B}_{i,k}) \doteq \mathrm{p}_{\text{int., glb.}}(\mathbf{B}_{i,k}) \not\equiv 1. \tag{4-10a}$$

Proof:
As long as *box context* is considered – for u-th ($u \in \overline{0, U}$) histogram bin b_u, k-th ($k \in \overline{0, m}$) box division and i-th ($i \in \overline{0, c_k}$) box $\cdot$:

$$x \in b_u \wedge \exists y \text{ in } \mathbf{B}_{i,k}.\, y \in b_u \implies x \in b_u. \tag{4-10b}$$

Equation (4-10b) means that in eq. (4-3c) may exist a case (or cases) of $x_{s,t,k} \overset{\text{def}}{=} x$ in which the following is true:

$$\neg\,(x \in b_u \wedge \exists y \text{ in } \mathbf{B}_{i,k}.\, y \in b_u) \wedge x \in b_u. \tag{4-10c}$$

Equivalently:

$$\forall y \text{ in } \mathbf{B}_{i,k}.\, y \notin b_u \wedge x \in b_u. \tag{4-10d}$$

Respectively:

$$\forall y \text{ in } \mathbf{B}_{i,k}.\, y \notin b_u. \tag{4-10e}$$

Such case(s) can possibly exist. Therefore, the following inequality may be stated as a conclusion drawn from eqs. (4-10c) to (4-10e) whose most clear illustration is possibly eq. (4-10c):

$$\mathbb{1}_{\{x:x\in b_u \wedge \exists y \text{ in } \mathbf{B}_{i,k}.y\in b_u\}}(x_{s,t,k}) \leqslant \mathbb{1}_{\{x:x\in b_u\}}(x_{s,t,k}). \tag{4-10f}$$

Thus:

$$\sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x:x\in b_u \wedge \exists y \text{ in } \mathbf{B}_{i,k}.y\in b_u\}}(x_{s,t,k}) \leqslant \sum_{t=1}^{\varepsilon^2} \mathbb{1}_{\{x:x\in b_u\}}(x_{s,t,k}) \equiv 1. \tag{4-10g}$$

Equivalently to eqs. (4-10c) to (4-10g), yet using simpler notation from eq. (4-3c) in place of the one from eq. (4-3b), existence of a case (or cases) is possible where $\left(\bigcup_{u=1}^{U} b_u\right) \cap b_{i,k} = b_{i,k} \subseteq \bigcup_{u=1}^{U} b_u$ is a *proper subset* of $\bigcup_{u=1}^{U} b_u$, i.e.:

$$\left(\bigcup_{u=1}^{U} b_u\right) \cap b_{i,k} = b_{i,k} \subsetneqq \bigcup_{u=1}^{U} b_u. \tag{4-11a}$$

Therefrom:

$$1 \not\equiv \mathbb{1}_{\{x:x\in(\bigcup_{u=1}^{U} b_u)\cap b_{i,k}\}}(x_{s,t,k}) \leqslant \sum_{u=1}^{U} \mathbb{1}_{\{x:x\in b_u\}}(x_{i,t,k}) \equiv 1; \tag{4-11b}$$

Therefore, basing on eq. (4-10g) or on eq. (4-11b), eq. (4-5d) from theorem 2 – – as well as eq. (4-9b) from theorem 6 – fails for *box-context global* probability in the intensity probability space $p_{\text{int.,\,glb.}}(\cdot)$:

$$p_{\text{int.}}(\mathbf{B}_{i,k}) \doteq p_{\text{int.,\,glb.}}(\mathbf{B}_{i,k}) \not\equiv 1. \tag{4-12}$$

The above eq. (4-12) proves the usability of $p_{\text{int.}}(\cdot)$. ■

4.6 CHAPTER SUMMARY

The considered *global* P-GBC variants are contrasted against their *local* counterparts in fig. 4.1 in both spatial as well as intensity probability spaces.

The performance of the global definitions presented above has been assessed in terms of FDE of a grayscale image, q.v. fig. 7.17. It has been found to be inferior when comparing the results of the algorithms utilizing the global definitions with their local-definitions-based counterparts. Therefore, this branch of methods has not been pursued further.

REMARKS ON THE ABBREVIATIONS OF THE NAMES OF METHODS Taking into account the abovementioned superiority of observed results of the *local* variants of the P-GBC, they are more extensively described and more often recalled in this thesis than their global counterparts. On this account as well as for the sake of notational clearness, PROBABILISTIC GRAYSCALE BOX-COUNTING based on image probability space: locally calculated variant (P-GBC-IMG-LOC) and PROBABILISTIC GRAYSCALE BOX-COUNTING based on intensity probability space: locally calculated variant (P-GBC-INT-LOC) are further often referred to as just PROBABILISTIC GRAYSCALE BOX-COUNTING based on image probability space (P-GBC-IMG) and PROBABILISTIC GRAYSCALE BOX-COUNTING based on intensity probability space (P-GBC-INT), respectively. When it does not clearly follows from the context whether the short term would refer to a whole sub-class of P-GBC-derived methods or just one of its local instances, its proper denotatum is stated explicitly. E.g., in the following sentence: "*P-GBC-IMG family of algorithms* encompasses

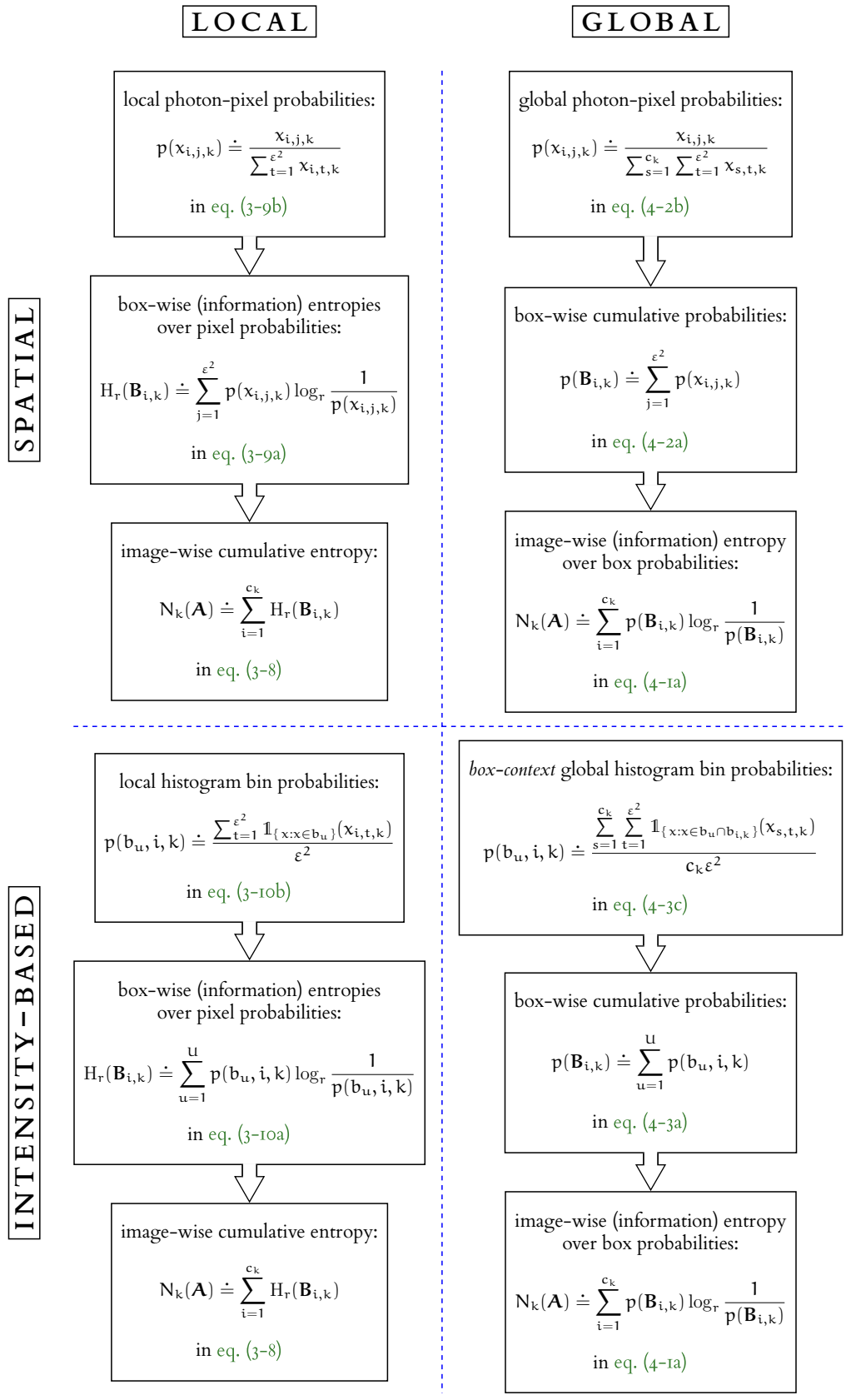


Figure 4.1: Flowchart comparison of P-GBC-IMG-GLB, P-GBC-INT-GLB, P-GBC-IMG--LOC and P-GBC-INT-LOC

P-GBC-IMG-LOC and P-GBC-IMG-GLB" it is clarified that the shortest abbreviation pertains to a category of methods. Complementarily, this phrase: "*P-GBC-IMG method* have performed better than P-GBC-IMG-GLB" refers to a comparison of two individual algorithms thus implicitly implying that the opposite context is assumed for the connoted name.

Part III

OTHER ASPECTS OF THE WORK RELATED TO FRACTAL DIMENSION ESTIMATION

The following part contains own contribution – a description of the algorithmic tools used to synthesize the artificial input data used for validation of the developed algorithms and their implementations. Importantly, the developments described in chapters 5 and 6 have served as well to facilitate selective and precise manipulations of various parameters of fractal- or fringe-like inputs. This has enabled the numerical in-silico research to be conducted with focusing it on properties of the investigated methods of FDE in relation to characteristics of the input data and their changes. The introduction of the synthetic input data has enabled the research to be conducted in a much more controlled manner than in case of restricting it to be based solely on the physically acquired images.

5 INPUT DATA SYNTHESIS USING RANDOM FRACTAL FUNCTIONS

5.1 PLANAR RANDOM FRACTAL FUNCTIONS BASED ON RESCALE-AND-ADD METHOD

Artificial data exhibiting fractal features that in particular may be parametrized in terms of fractal dimension (FD) and lacunarity – a measure quantifying *heterogeneity* and overall *gappiness* or *rotational invariance* of fractals [82] – have been generated using *rescale-and-add* (*RAA*) method developed by Dietmar Saupe in 1989 and described in detail in [81]. This method allows for point or global evaluation of *multi-variable random fractal functions* (*RFFs*) and has been proven as superior to some of its popular counterparts in terms of parametrization possibilities (as it offers two fractal-related parameters and allows for imposing spatial locality on both of these parameters in a simple manner), capability for parallelization and – in many cases – computational complexity. The actual algorithm used for random fractal data generation is author's own implementation of two-dimensional (2-D) exemplification of an abstract n-D concept of RAA approach. The obtained data are in the form of 8-bit grayscale PNG images (with no transparency) of size 512 px × 512 px that corresponds with the format of the spatially cropped lateral shearing interferometry (LSI) frames used for FRACTAL DIMENSION estimation (FDE) algorithms comparison on real data. Artificially generated images allow for performing analogous comparison on synthetic data using the full spectrum of FD values from one to two with arbitrarily fine steps. Such comparison has higher plausibility in the axis of reference FD values, i.e., the abscissæ, as they are not estimated by binary box-counting (BBC) algorithm or in any other way. Contrarily, they are given as exact input to the procedure synthesizing artificial fractals that are subsequently feed into estimating algorithms that are to be compared. This provides i.a., additional validation for FDE algorithms and their implementations.

Mathematical content of this chapter is partly based on [81].

Definition 3. Two-dimensional *random fractal function* via *rescale-and-add* method:

Let $S_2 : \mathbb{R}^2 \to \mathbb{R}$ be a real function that has values at integer lattice points of $\mathbb{R}^2$ defined by Gaussian random variables of zero mean and the same variance at all points. Further, let $S_2(x, y)$ be a smooth interpolation from the data at the integer lattice points. Then, the function $\tilde{V}_2 : \mathbb{R}^2 \to \mathbb{R}$ defined by:

$$\tilde{V}_2(x, y) \doteq \sum_{k=k_0}^{\infty} \frac{1}{r^{kH}} S_2(r^k x, r^k y), \tag{5-1}$$

with $r > 1$ determining *lacunarity*, $0 \leqslant H \leqslant 1$ and $k_0 \leqslant 0$, is a *random fractal function* whose graph has *fractal dimension* equal to $\tilde{D}$:

$$\tilde{D} = 2 + 1 - H = 3 - H. \tag{5-2}$$

□

In practical applications, limits of the summation in eq. (5-1) within definition 3 have to be finite and defined in a way to credibly and effectively approximate $V_2(\cdot, \cdot)$. Also, $\tilde{D}$ mentioned in definition 3 is based upon an interpretation of FD characterized by three-dimensional (3-D) spatio-intense image space. Thus, $\tilde{D}$ can vary from two to three. However, in this thesis another interpretation is used, where FD is calculated over images embedded in 2-D space that have purely spatial coördinates and intensity being regarded separately in different manner; intuitively, intensity values may be understood here as a kind of weight. Thus, FD regarded further and denoted as D has to be reduced by one relatively to $\tilde{D}$ in order to be consistent with the rest of the work:

$$D \doteq \tilde{D} - 1 = 2 - H. \tag{5-3}$$

Regarding the considerations above and algorithmic remarks given in literature [81], the actual implementation has been based on the following: when 2-D synthetic grayscale image of size 512 px × 512 px depicting *RFF* is to be generated using the *RAA* method with FD given by $D \in [0, 2]$ and lacunarity determined by factor $r > 1$, it may be approximated by:

$$V_2(x, y) \doteq \sum_{k=k_0-k_{0m}}^{k_1+k_{1m}} \frac{1}{r^{kH}} S_2(r^k x, r^k y), \tag{5-4a}$$

where:

$$r \in \mathrm{set}\sqrt{2}, 2, 4 \tag{5-4b}$$

is the lacunarity parameter used in this thesis as well as in [81];

$$H \doteq 2 - D \in [0, 1] \tag{5-4c}$$

may be interpreted as Hölder exponent in a squared version of Hölder condition;

$$k_0 \doteq -\frac{\log_2 L}{\log_2 r} \tag{5-4d}$$

is the left-side end of summation range asserting that the dominant (lowest) spatial frequencies $r_{k_0}/2\pi$ of the auxiliary function $S_2(\cdot, \cdot)$ are smaller than the inverse of largest scale L of objects in the image:

$$L \doteq 512 \tag{5-4e}$$

is the largest scale of objects in the image corresponding to 512 px × 512 px image side lengths;

$$k_{0m} \doteq 2 \tag{5-4f}$$

is the left-side margin for summation range enhancing the lowest frequencies condition;

$$k_1 \doteq \frac{\log_2 2/\Delta}{\log_2 r} \tag{5-4g}$$

is the right-side end of summation range asserting that the highest frequencies $r_{k_1}/2\pi$ of the auxiliary function $S_2(\cdot, \cdot)$ are smaller than the Nyqyist limit imposed by the sampling distance (spatial resolution) Δ:

$$\Delta \doteq 1 \tag{5-4h}$$

is the sampling distance corresponding to 1 px spatial resolution);

$$k_{1m} \doteq 8 \tag{5-4i}$$

is the left-side margin for summation range asserting that a few additional summation terms of frequencies above the upper limit are not clamped, as for small values of H they tend to have amplitudes $\frac{1}{r^{kH}}$ of the auxiliary function $S_2(\cdot, \cdot)$ large enough to be considered non-negligible when intending to achieve æsthetically pleasing results – as this aspect is emphasized in [81], k_{1m} is set to a value noticeably higher than k_{0m} in the developed implementation;

$$\begin{aligned} S_2(x, y) \doteq\ & s_x \cdot s_y \cdot V(\lfloor x + 1 \rfloor, \lfloor y + 1 \rfloor) + \\ & + (1 - s_x) \cdot s_y \cdot V(\lfloor x \rfloor, \lfloor y + 1 \rfloor) + \\ & + s_x \cdot (1 - s_y) \cdot V(\lfloor x + 1 \rfloor, \lfloor y \rfloor) + \\ & + (1 - s_x) \cdot (1 - s_y) \cdot V(\lfloor x \rfloor, \lfloor y \rfloor) \end{aligned} \tag{5-4j}$$

is the auxiliary, differentiable, interpolating function, $\lfloor x \rfloor$, $\lfloor y \rfloor$ are integer parts of x and y, respectively, and:

$$\begin{cases} s_x \doteq \{x\}^2 (3 - 2\{x\}) \\ s_y \doteq \{y\}^2 (3 - 2\{y\}) \end{cases}, \tag{5-4k}$$

where:

$$\begin{cases} \{x\} \doteq x - \lfloor x \rfloor \\ \{y\} \doteq y - \lfloor y \rfloor \end{cases} \tag{5-4l}$$

are the non-negative fractional parts of x and y, respectively;

$$V(k, l) \doteq t_{(k-1 \bmod N)+1, (l-1 \bmod N)+1} \tag{5-4m}$$

is a function simulating infinite 2-D array of random numbers attached to lattice points with integer coördinates (k, l). $V(\cdot, \cdot)$ is calculated using a finite table of N^2 random numbers, where (N, N) is the spatial period of $S_2(\cdot, \cdot)$ and N is usually equal to 50 or 100 [81] – in the developed implementation $N \doteq 100$. This table stores the elements of matrix $\mathbf{T}$:

$$\mathbf{T} \doteq (t_{i,j})_{N \times N}, \tag{5-4n}$$

with $t_{i,j} (i, j \in \overline{1, N})$ being random variates (or realizations of independent and identically distributed random variables) of *normal random variable* T with *zero mean*:

$$T \sim \mathcal{N}(0, \sigma^2) \tag{5-4o}$$

which in the developed implementation has the *standard normal distribution*:

$$T \sim \mathcal{N}(0,\, 1). \tag{5-4p}$$

In particular, for $L \doteq 512 = 2^9$ and $\Delta \doteq 1$, and for the most homogeneous fractal, i.e., of lowest lacunarity, among exemplifications considered in [81] – with lacunarity parameter $r \doteq \sqrt{2} = 2^{\frac{1}{2}}$:

$$\begin{aligned} k_0 - k_{0m} &\doteq -\frac{\log_2 L}{\log_2 r} - 2 \doteq -\frac{\log_2 2^9}{\log_2 2^{\frac{1}{2}}} - 2 = -\frac{9\log_2 2}{\frac{1}{2}\log_2 2} - 2 = \\ &= -18 - 2 = -20, \end{aligned} \tag{5-5a}$$

$$\begin{aligned} k_1 + k_{1m} &\doteq \frac{\log_2 2/\Delta}{\log_2 r} + 8 \doteq \frac{\log_2 2/1}{\log_2 2^{\frac{1}{2}}} + 8 = \frac{\log_2 2}{\frac{1}{2}\log_2 2} + 8 = \\ &= 2 + 8 = 10, \end{aligned} \tag{5-5b}$$

therefore in that case there are $(k_1 + k_{1m}) - (k_0 - k_{0m}) + 1 \doteq 10 - (-20) + 1 = 31$ summands in eq. (5-4a).

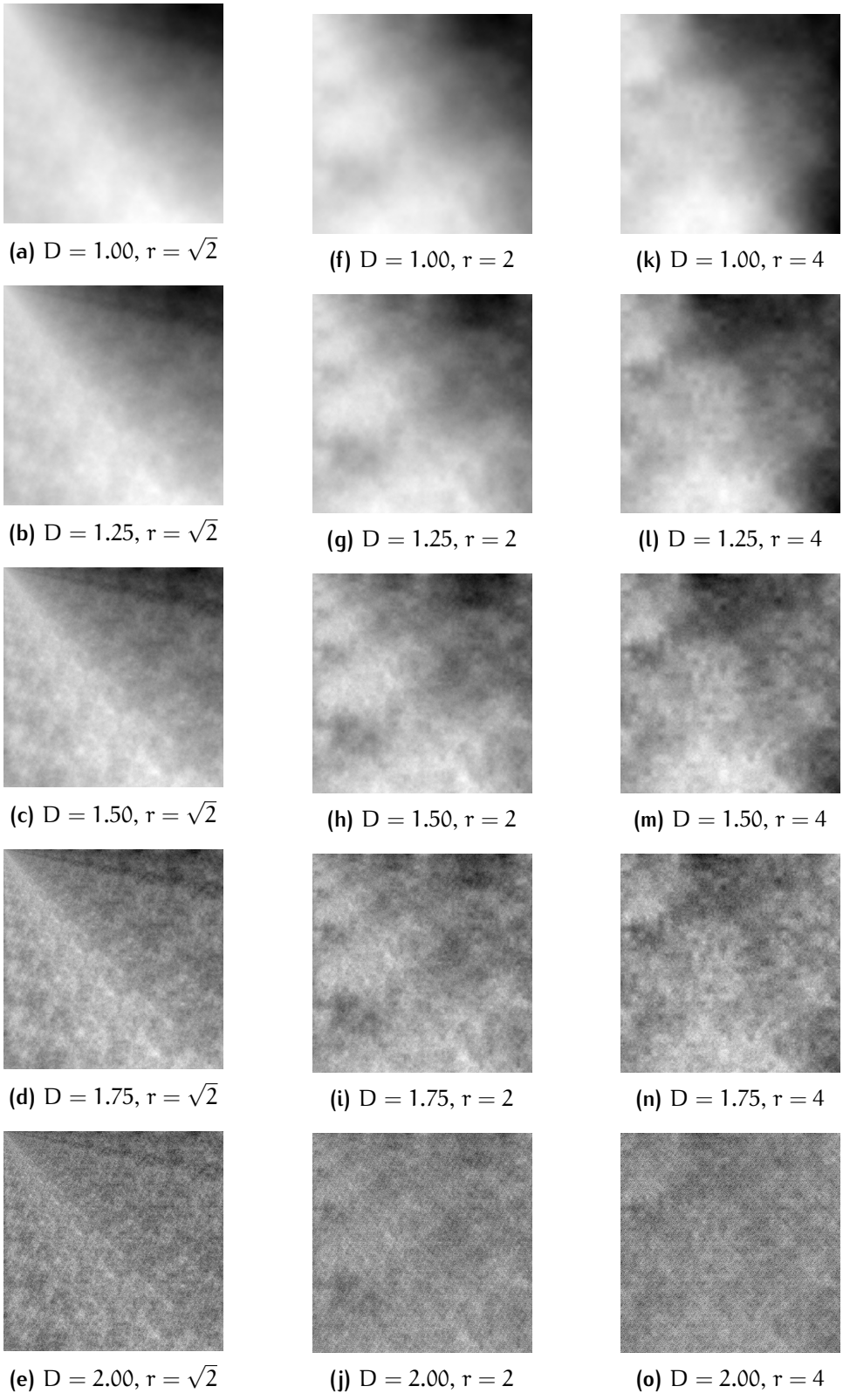
The implementation developed in MATLAB numerical computing integrated development environment (IDE) allows for parallelized computations. In the investigated cases of 2-D RFFs images of size 512 px × 512 px, such parallelization vastly improves calculation speed and thus reduces average image synthesis time. The loop-based variant has primarily been timed to execute for about 20 000 s, i.e., computations have lasted more than five and a half hours. When the parallelization has been programatically facilitated, this time has been reduced to about 0.6367 s, i.e., computations have been completed in just over half a second. The achieved change in computational efficiency has reached four to five orders of magnitude. The mentioned improvement in computational load has been accomplished by designing the algorithm to perform some critical operations in a matrix-wise manner in 2-D. These modifications have allowed for the usage of MATLAB's built-in array optimizations. Functionally, when calling the implemented random fractal images synthesis function declared as `function V_2 = raa2(x, y, varargin)`, q.v. listing C.4, this parallelized mode is invoked when the coördinates of the point in which RFF is to be calculated are entered not as single numbers `x` and `y` representing indices x and y, respectively, but rather as index ranges. Particularly, to compute the whole L px × L px image as single matrix, `x` and `y` coördinates are to be given as vectors of index values of interests, preferably sorted increasingly. E.g., column and row vectors may be used, respectively, i.e., $x \doteq [1\ \ 2\ \ \cdots\ \ L]^T$ and $y \doteq [1\ \ 2\ \ \cdots\ \ L]$.

Figure 5.1 presents illustrative synthetic fractal images depicting approximations of *RFFs* generated using the *RAA* method proposed by Saupe [81], as presented in definition 3 and by implementation based on eq. (5-4). FD values used for research range from one to two in steps of 0.05, of which one fifth have been chosen to be visualised in fig. 5.1, where the step of presented FD values is thus equal to 0.25. Similarly, three levels of lacunarity [82] have been used for experiments, with values of lacunarity parameter r being as suggested in the original research by Saupe [81], i.e., equal to $\sqrt{2}$, two and four. Visual effect that the influence of all of these three r values has on the resulting fractal-approximating images is also presented in a column-wise manner in fig. 5.1.

Figure 5.2 depicts FD estimates compared for synthetic data of Saupe's RAA algorithm for evaluating multi-variable RFFs computed using four algorithms: probabilistic grayscale box-counting based on image probability space (P-GBC-IMG), probabilistic grayscale box-counting based on intensity probability space (P-GBC-INT), weighted grayscale box-counting (W-GBC) and differential box-counting (DBC) algortihms. As stated above, FD values used for this experiment range from one to two in steps of 0.05, forming three color-marked families of plots that correspond to three values of lacunarity parameter r ($\sqrt{2}$, two and four) used. These altogether comprise twelve plots presented in the figure.

Figure 5.3 shows plots extending fig. 5.2 by presenting two another variants of cut-off applied on extreme data points during linear regression stage of common box-counting (BC) procedure for all grayscale BOX-COUNTING (GBC)-based algorithms. The default BC cut-offs have been set to zero in application to Saupe's RAA--based random fractals – this is presented by fig. 5.2. In the original algorithm desing, the numbers of elements to be cut off on both sides have been non-zero. These numbers have been reduced in the descripted case, as it has led to an increase in correlation with Saupe's D as well as due to the fact that with the original number of cut-off points equal to three and one for lower and upper cut-off, respectively, P-GBC-INT estimates have been declining for Saupe's D values above about 1.65. The original cut-off values have been optimized for simple synthetic images and real (LSI-based) data regarding boundary effects as well as computational complexity, q.v. chapter 8. This rendered to be sub-optimal for Saupe's RAA RFF-based images by causing the aforementioned ambiguity for P-GBC-INT, as for these cut-off thresholds $\mathrm{FD}_{\text{P-GBC-INT}}(\mathrm{D})$ function has not been injective. Consequently, a reduction in cut-off thresholds has mitigated this issue for probabilistic GRAYSCALE BOX-COUNTING (P-GBC)-based algorithms. Nevertheless, at the same time it introduced a problem of a similar kind for W-GBC algorithm. This is noticeable

when comparing fig. 5.3b to fig. 5.2 where its indication is an increase in correlation coefficient ρ for both P-GBC-based algorithms as well as a decrease for W-GBC algorithm. The latter change is especially profound for the highest used lacunarity parameter r value of four. This altogether suggests that the optimal values of parameters for a BC-based algorithm are strongly dependent by both the input image class and size as well as on the chosen method variant itself. Moreover, for the purpose of assessing the overall applicability of methods, it is worth noting that for the majority of possible values of cut-off thresholds (of which only a small part is presented in fig. 5.3, but many more have been investigated), the values of estimators may significantly change. However, their general dynamics remains relatively unchanged. Thus, a conclusion may be drawn that cut-offs applied during linear regression at BC stage are of vital importance for the fine tuning of the performance of the algorithm on a specific class of input data, nevertheless they are not critical for some more general conclusions about its properties. E.g., many different cut-offs may be chosen to conclusively determine if a given GBC-based algorithm is applicable for classification tasks as well as if sensitivity and variability characteristics of its outputs are consistent with analogous results obtained for other FD estimators.



(a) $D = 1.00$, $r = \sqrt{2}$ **(f)** $D = 1.00$, $r = 2$ **(k)** $D = 1.00$, $r = 4$

(b) $D = 1.25$, $r = \sqrt{2}$ **(g)** $D = 1.25$, $r = 2$ **(l)** $D = 1.25$, $r = 4$

(c) $D = 1.50$, $r = \sqrt{2}$ **(h)** $D = 1.50$, $r = 2$ **(m)** $D = 1.50$, $r = 4$

(d) $D = 1.75$, $r = \sqrt{2}$ **(i)** $D = 1.75$, $r = 2$ **(n)** $D = 1.75$, $r = 4$

(e) $D = 2.00$, $r = \sqrt{2}$ **(j)** $D = 2.00$, $r = 2$ **(o)** $D = 2.00$, $r = 4$

Figure 5.1: Illustrative grayscale visualizations of two-variable RFFs of size 512 px × 512 px generated using author's implementation of the *rescale-and-add* method [81] with FD D varying row-wise in quarter steps from one to two and lacunarity parameter r varying column-wise in range $\{\sqrt{2}, 2, 4\}$

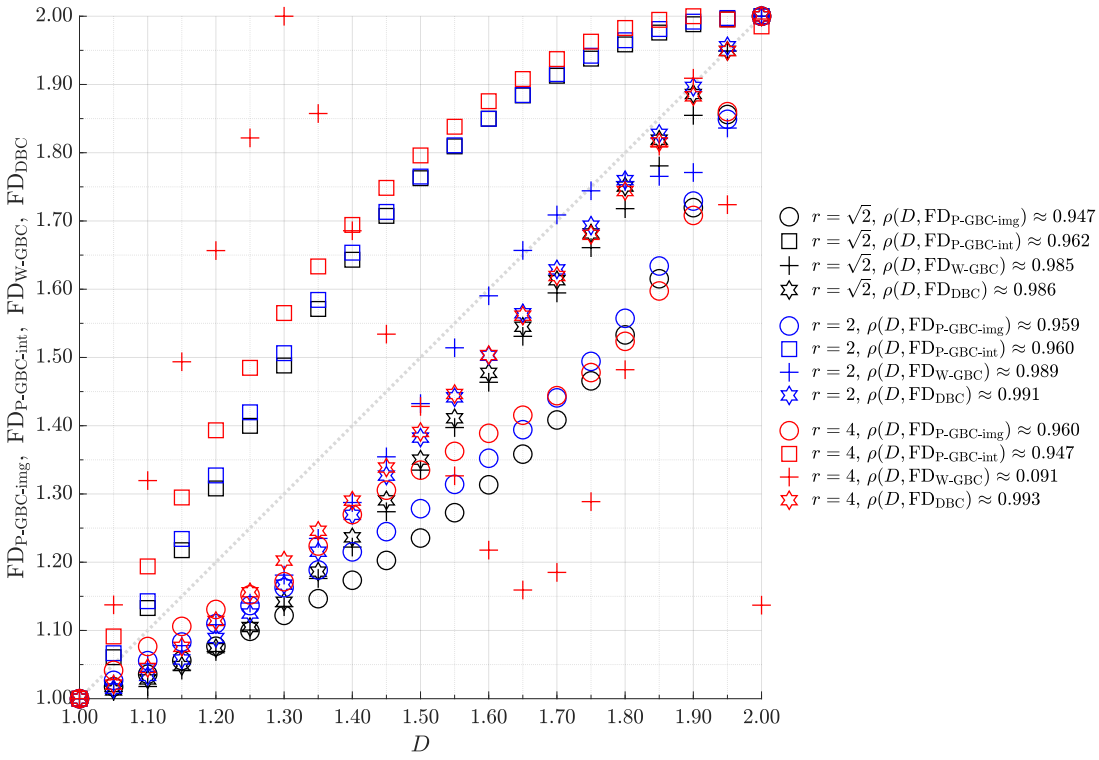
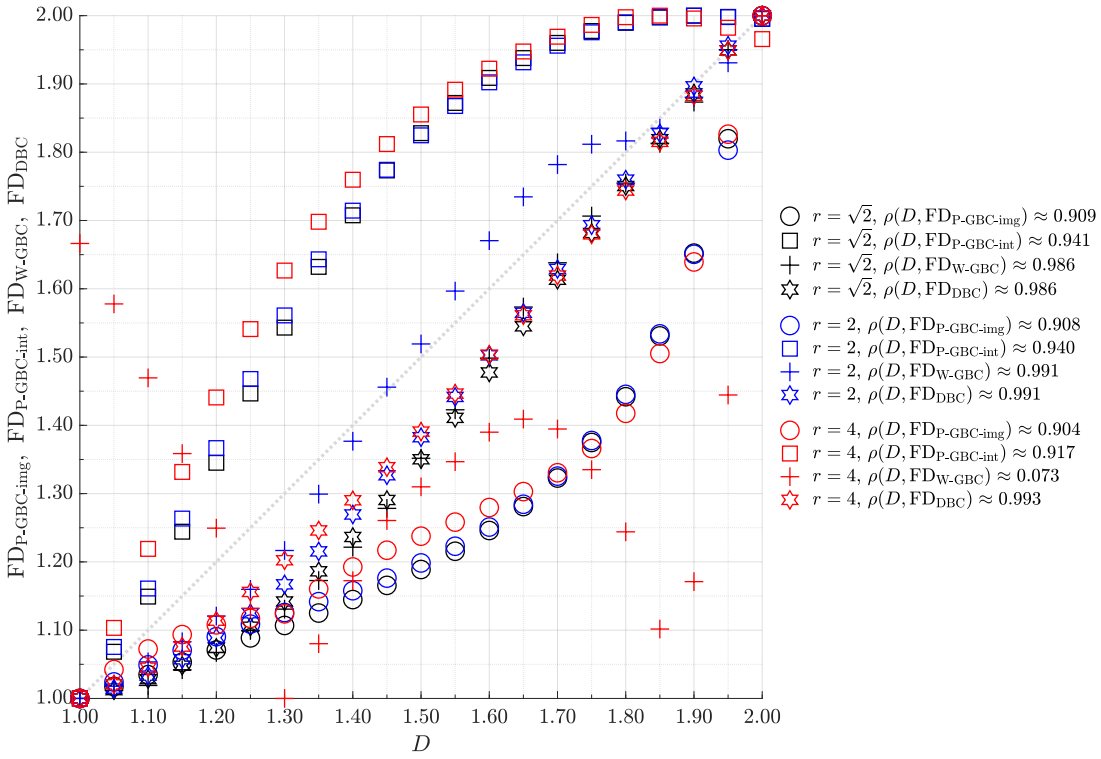
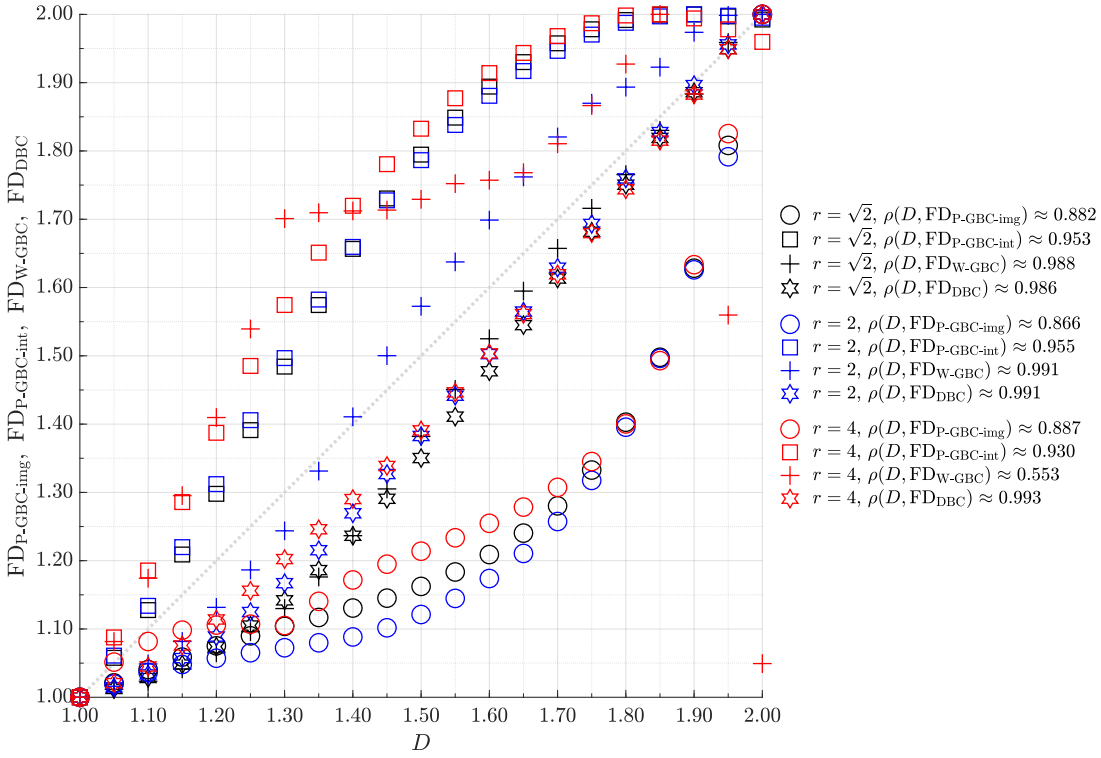


Figure 5.2: Comparison of FD estimates for synthetic data of Saupe's RAA algorithm for evaluating multi-variable RFFs computed using $FD_{P\text{-}GBC\text{-}IMG}$, $FD_{P\text{-}GBC\text{-}INT}$, $FD_{W\text{-}GBC}$ and FD_{DBC} estimators without cut-off of extreme points in GBC-based algorithms during linear regression of $N_\varepsilon(\varepsilon)$ data performed by core BC procedure. There are three families of plots indicated by black, blue and red markers, corresponding to lacunarity parameter r set to $\sqrt{2}$, two and four, respectively. Plotted values are normalized by adaptive linear rescaling of ordinates to match their minima and maxima with abscissæ and to assert positive direction of output variation



(a) First and last BC regression data points cut-off



(b) Two first and two last BC regression data points cut-off

Figure 5.3: An extension to fig. 5.2 showing two other cut-off variants of extrema in GBC--based algorithms during linear regression of $\mathrm{N}_\varepsilon(\varepsilon)$ data. The adaptive rescaling of plots and the meaning of colors are identical to these of fig. 5.2

5.2 ANOTHER APPROACHES TO SYNTHESIS OF FRACTAL IMAGES OF A VARIABLE DIMENSION

A common and simple approach to providing synthetic fractal for the input data set is to utilize Gaussian noise. Sometimes it is a zero-mean noise with varying standard deviation that is added to a surface at a constant gray level having the middle value of the intensity range of the used image type and with clipping applied at the intensity range limits [17, pp. 3 (117) – 4 (118)]. FD of images generated in such way increases with the variation of the noise. In 3-D interpretation of the images, it varies from exactly two to asymptotically three. Nevertheless, this method is also charged with serious drawbacks, e.g., the dependence of FD on the standard deviation is strongly non-linear, q.v. [17, fig. 3]. Moreover, the upper value is never actually reached and the lower one does not have a universally clear meaning of the images. E.g., for 2-D interpretation such a constant surface of a half-background and half-foreground level may be interpreted as having intermediate value of FD – equal to one. This would be the case for some weighted approaches related to W-GBC. When examined with other implementations it may be interpreted as a completely flat object that due to lack of variation of intensity has FD value of zero. This would be the case for DBC-or P-GBC--INT-based algorithms. Finally, such an object that has the size of the whole image and is of a constantly non-background value, due to its planar nature or 2-D shape may be interpreted as having FD value of two or close to two. This would be the case for some methods operating in image probability space and related to P-GBC-IMG. Moreover, Gaussian solution in its simplest variant does not allow for controlling other parameters of the produced images, e.g., lacunarity. Therefore, for advanced applications it may be more suitable to exploit more elaborated solutions for synthesis of fractal images. These may include or relate to randomization of deterministic fractals, percolations, Brownian motion, Gaussian randomness, mean square displacement and *random midpoint displacement*, sometimes also referred to as Peitgen's approach [11], although it has been introduced earlier by Alain Fournier, Donald S. Fussell, and Loren C. Carpenter [83]. Relatively recently, Peitgen's approach has also been extended to generation of color fractal images [44], q.v. section 1.2.3. Notwithstanding, none of these approaches is completely free of imperfections in comparison with Saupe's concept. Also in use are methods that involve Mandelbrot's-Weierstrass' functions or Perlin's turbulence functions. A composite approach to combine advantages of both methods into a single one

which would also alleviate their shortcomings and limitations has led to formulation of one of most robust techniques available. This research has been conducted by Dietmar Saupe and has resulted in introduction of RAA method published in 1989 [81]. Apart from purely fractal-oriented research, this approach is to date also heavily exploited in texturing and modeling, especially in tasks related to terrain surface simulations [84, 85], but also for generation of cloud textures [86] etc.

6 ARTIFICIAL FRINGE PATTERN SIMULATION

A method for generating artificial images of fringe patterns has been developed and the synthesized data has been used for validation and test purposes, e.g., simulation of interferograms and their phase disruptions. Artificially constructed data simulating interference images acquired in conditions entailing the presence of fringe phase disruptions in the patterns have been used for assesing fractal dimension estimation (FDE) reliablity as well as algorithms validation and parameters fine-tuning. Illustrative visualizations of such artificial interferometric data – in this case affected by sinusoidal disruption of fringe pattern phase – are presented for comparison in fig. 6.2.

6.1 GENERATING SYNTHETIC IMAGES

6.1.1 Fringe pattern synthesis for space-wise analysis

To enable validation and additional experiments at various stages of the processing, a method for generating artificial interferograms has been designed and implemented. Synthetic images might be generated with a given frequency (preferably mimicking the typical of lateral shearing interferometry (LSI), that in this case usually oscillates around one seventh of the Nyquist's frequency), orientation, phase disturbance level (sinusoidal or Gaussian), resolution and dimensions. An illustrative result is depicted by fig. 6.1, where artificially generated simulation depicts a central Gaussian phase distortion.

An auxiliary research has been conducted for one of tear film surface quality (TFSQ) estimators (FD_{BBC}) and its variability in terms of the frequency changes. Also, sinusoidal phase disturbance level has been investigated.

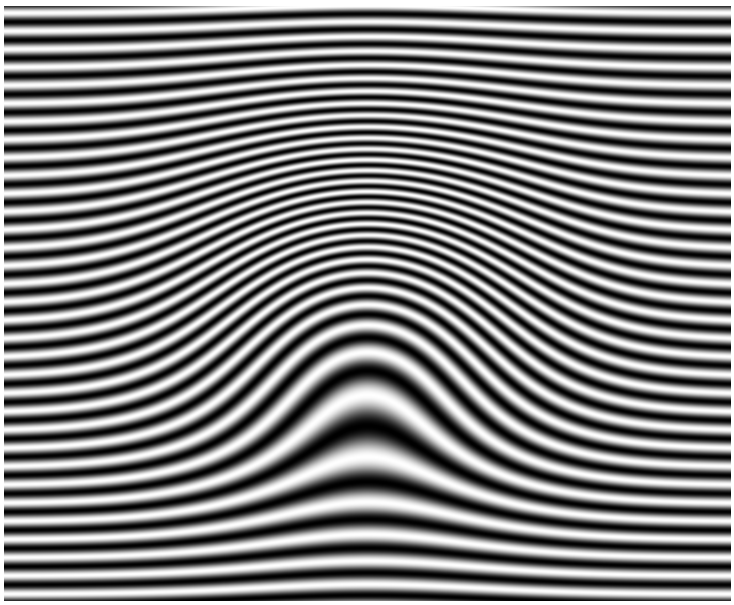


Figure 6.1: Artificially generated image of interferometry pattern – a simulation of human pre-corneal tear film LSI imagery

6.1.2 Fringe pattern synthesis for supplementary spatial-frequency-wise research

A synthetic interference pattern has been rendered. Its frequency and degree of structural disruption have been subject to alterations during the measurements using the following parameters:

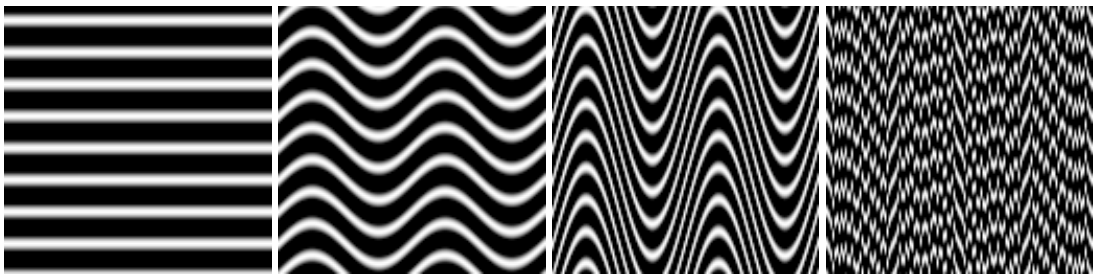
(1) fringe pattern orientation (expressed as the direction of the instantaneous phase variability): $\pi/2$;
(2) image space coördinates range: $[-1, 1]$ in both horizontal and vertical axis;
(3) image resolution: $128\,\mathrm{px} \times 128\,\mathrm{px}$ – a value chosen to be the same as size of the subframe utilized by the algorithm calculating TFSQ descriptor values for the real data;
(4) fringe spatial frequency range (in the variability direction): $[0, f_N]$ linearly distributed into a 100-element vector, where $f_N = \frac{f_s}{2}$ is the Nyquist's frequency being half of the sampling frequency for the image of a given resolution within a given range of coördinates;
(5) the degree of a fringe structure disruptions – expressed as the amplitudes (in radians) of the sinusoidal phase noise with a wavelength equal to 1 – used for generation of a graphs family to be plotted included: 0, 4, 15, 110, 275, 500 and 10 000 with the first four classified later in terms of convergence as being

from low to moderate and last two being large distortions with one intermediate value;

(6) orientation deviation used for each fringe pattern to average fractal dimension (FD) estimator values and thus mitigate numerical effects introducing additional variance or bias into the estimation due to, e.g., moiré-like phenomena (that has been observed to occur and have strong influence on graphs generated without averaging): $-5°$ to $5°$ in equal $0.5°$ steps (resulting in 21 measurements per one data point).

The exact way of the patterns generation involved some additional trigonometrical calculations. To check their form please refer to listing listing C.3 which include key lines of code utilized by the procedure along with the comments explaining their significance. In the key fragments of the lsiting a supplementary algorithm to generate synthetic interferogram image with or without sinusoidal fringes phase disruptions is presented. The code also serves for the purpose of testing the dependency between fringes frequency and the esitmate of FD of the pattern as well as the degree of disturbance and FD.

In four groups containing eight illustrative images in fig. 6.2, there are fringe patterns depicted that involve sinusoidal phase modulations to simulate interferogram disruptions. There are both grayscale and binarised versions (different FD estimators make use of either monochrome or grayscale inputs) of data with: zero, slight, moderate and big perturbations, i.e., disturbances having phase shift amplitudes of, respectively, 0, 4, 15 and 500. The described images represent fringe patterns having spatial frequency of $\frac{f_N}{7}$, that for the image coördinates range and resolution being considered corresponds to single fringe spatial wavelength equal to about 14.3 px. This value has been chosen due to its compatibility with fringe structures typically occurring within records of real data acquired in an in-vivo fashion.



(a) Undisrupted: $A_\varphi = 0$ **(b)** Visible disturbance: $A_\varphi = 4$ **(c)** Medium disturbance: $A_\varphi = 15$ **(d)** Severe disturbance: $A_\varphi = 500$

(e) No disruption: $A_\varphi = 0$ (binarised) **(f)** Visible disturbance: $A_\varphi = 4$ (binarised) **(g)** Medium disturbance: $A_\varphi = 15$ (binarised) **(h)** Severe disturbance: $A_\varphi = 500$ (binarised)

Figure 6.2: Illustrative simulated image of fringes with different levels of sinusoidal phase disruption – in grayscale (upper row) and after binarisation (bottom row). Spatial frequency of fringes is fixed (with the value of roughly one seventh of the Nyquist's frequency). Phase disruption level increases from left (ideal uninterrupted case) to right (significantly disturbed fringe pattern). A_φ denotes the amplitude of the phase disturbance (in radians)

Part IV

PRACTICAL ASPECTS OF FRACTAL DIMENSION ESTIMATION FOR GRAYSCALE IMAGES

The following part contains original contribution – it presents the results of the conducted research.

7 MATERIALS

7.1 LATERAL SHEARING INTERFEROMETRY IMAGERY*

Interferometric images have been exploited in numerous scientific and industrial applications. Among them, surface roughness examination and defect detection in thin layers, such as human tear film surface quality (TFSQ) assessment, constitutes a matter of continuously growing interest. Namely, in biomedical research studies it increases its prevalence due to significance in non-invasive diagnostic methods and accuracy [87, 88].

In recent years, efforts have been undertaken to perform thorough interferometric image feature analysis by classical morphological image processing or by switching to frequential spectrum prior to performing the actual data analysis. Similarly, attempts have been made to interpret interferograms as patterns, e.g., via gray--level co-occurrence matrix (GLCM)-based texture analysis [89], and to determine their statistical parameters to qualitatively or quantitatively assess the useful information contained within the image data [72, 90]. There have been performed research works regarding only one of these approaches, e.g., frequency-spectrum-wise analysis [73, 91, 92] or statistical texture analysis [93]. Also in some cases both of them have been compared [94] in terms of efficacy or robustness for interferograms [95] and, in general, medical imaging data [96, 97]. To date, however, the utilization of fractal dimension estimation (FDE) to lateral shearing interferometry (LSI) image analysis and classification has neither been employed with practical applications in empirical manner, nor systematically studied theoretically, even though it tends to be widely tested approach in related medical fields [38, 98] and other applications demanding three-dimensional (3-D) shape features estimation of shaded or textured surfaces [20]. The aim of this thesis has been to perform such a fractal feature analysis and to preliminarily evaluate its applicability, benefits over concurrent methods that can be achieved with their utilization to this field and, finally, to identify potential pitfalls. These may occur when practical employment of the method would be considered, especially in context of data gathered in an in-vivo manner,

* This section is based on author's paper published in *Applied Optics* [74].

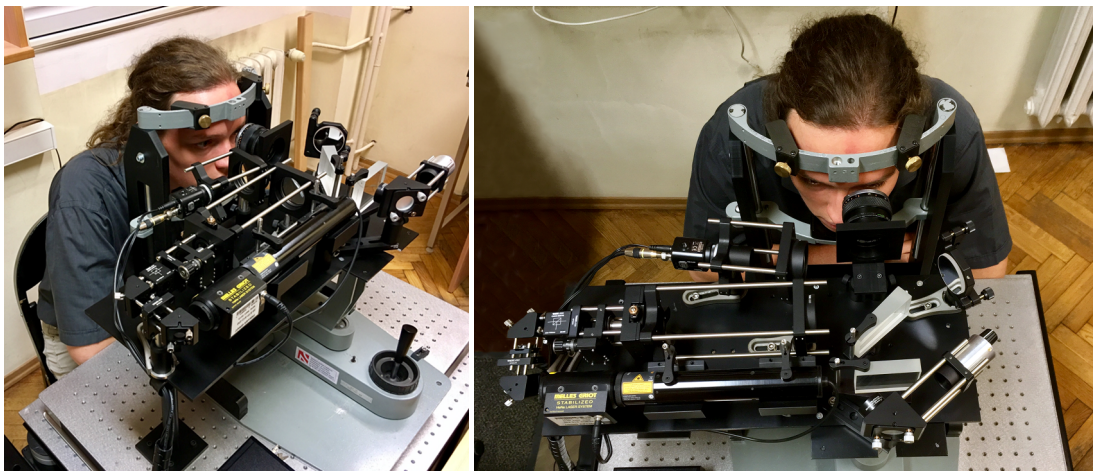


Figure 7.1: The hardware setup of LSI imaging system

while images are acquired under conditions of restricted control and may be subject to considerable disturbances.

The section structure is structured as follows: section 7.1.1 presents the research methodology, section 7.1.2 is focused on the results and their statistical analysis while section 7.1.3 describes the conclusions and future recommendations.

7.1.1 Research methodology

Experimental setup

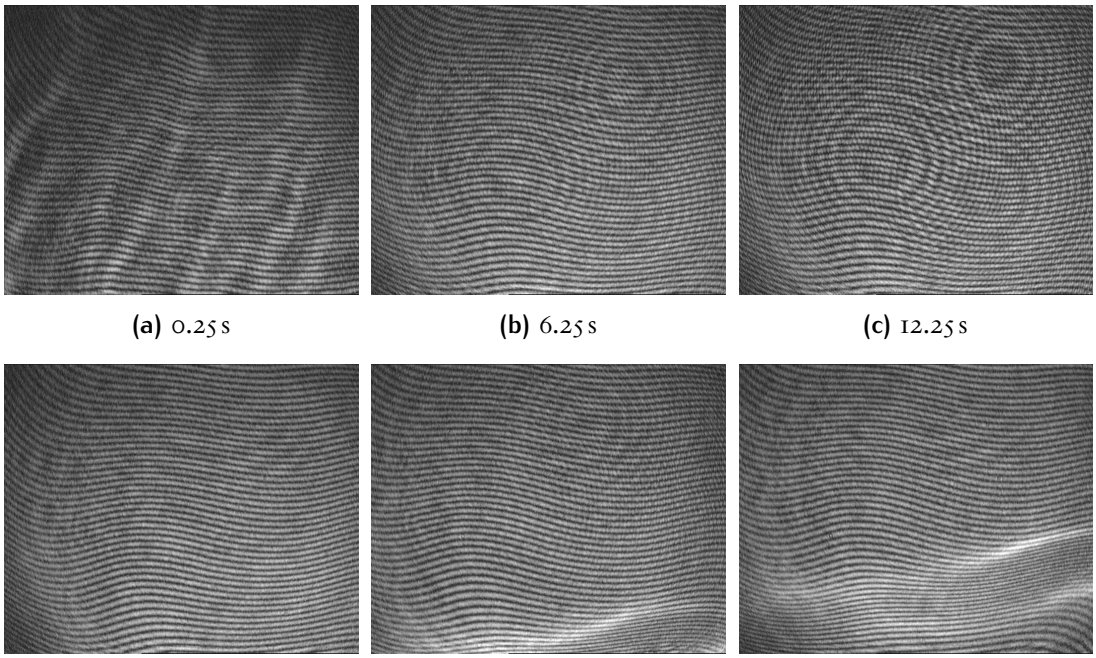
Experimental data have been collected by in-vivo means using the interferometric device with lateral shearing configuration. The most important hardware parts of LSI apparatus, cf. fig. 7.1, include: a He-Ne laser as the light source, mirrors reflecting the beam, a collimator to expand beam to the desired width of approximately 30 mm [99], a beam splitter (8/92) directing the beam towards the eye, a lens focalizing the beam on the cornea and a optical wedge distributing the light into two near-parallel beams (after reflection on the pre-corneal tear film) and a charge-coupled device (CCD) video camera.

LATERAL SHEARING INTERFEROMETRY DATA ACQUISITION Data acquired with LSI apparatus have been gathered for two groups of human subjects: 69 healthy patients and 41 diagnosed as suffering from dry eye syndrome (DES). Measurements have been performed under suppressed blinking condition (SBC), i.e., the examined subjects have been instructed to blink a few times and focus their sight on the apparatus fixation target before measurements start. Subsequently, they have been asked

to hold their eyes open for as long as possible, but for no longer than 30 s (maximum recording period for LSI sequences) [73, 91].

Used for the research have been 110 uncompressed audio video interleave (AVI) files, recorded in the resolution of 704 px × 576 px at 25 fps.

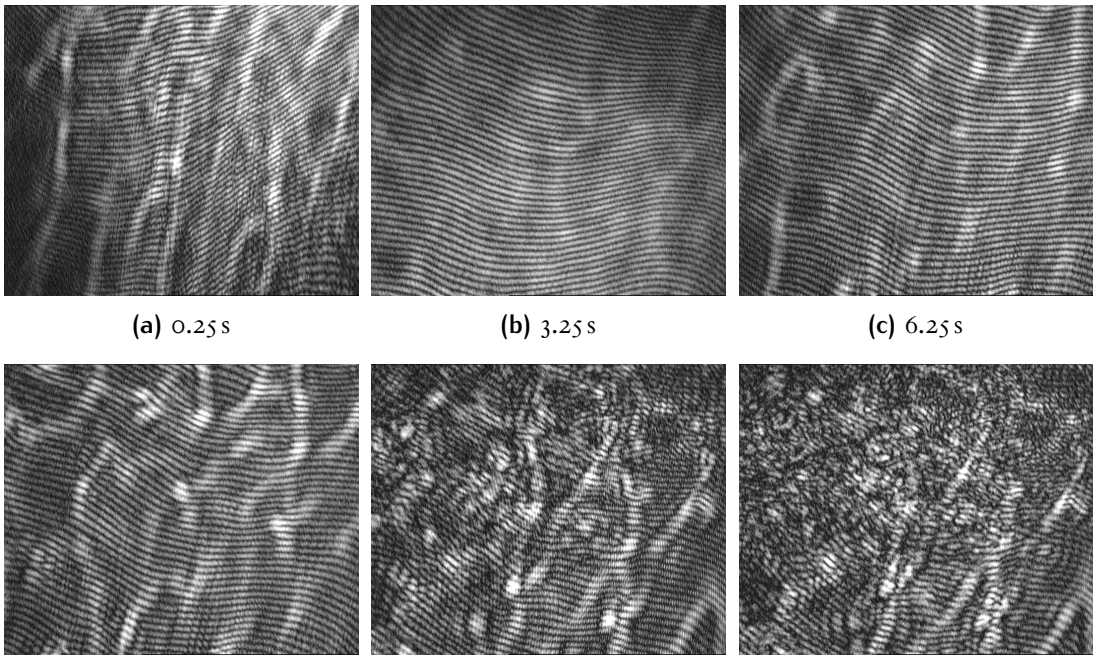
Twelve chosen instances of still LSI images – extracted from AVI sequences – that have been recorded by the measurement unit are shown in figs. 7.2 and 7.3. Subjects that have been examined to obtain the LSI data have also been assessed for DES using standard clinical measures independently of the interferometric imaging. In a typical case of the resulting set of interferograms, after the stabilization period of the tear film, the fringes' disturbance is noticeably smaller for a healthy patient (e.g., fig. 7.3c) than for a dry eye syndrome-affected one (e.g., fig. 7.2b). Also, the interblink time necessary for a complete deterioration of the tear film tends to be significantly shorter for DES patients class, e.g., 13.44 s in fig. 7.3f (where tear film condition has been so poor that it has forced the patient to blink and the recording has been terminated shortly after) vs 29.80 s in fig. 7.2f where tear film has not been substanitally deteriorated even though the maximum recording time has been reached, as recordings have been manually stopped after about 30 s, cf. section 7.1.1 and [73].



(a) 0.25 s **(b)** 6.25 s **(c)** 12.25 s

(d) 18.25 s **(e)** 24.25 s **(f)** 29.80 s

Figure 7.2: LSI – typical images for healthy human pre-corneal tear film shown with recording time after blink



(a) 0.25 s **(b)** 3.25 s **(c)** 6.25 s

(d) 9.25 s **(e)** 11.92 s **(f)** 13.44 s

Figure 7.3: LSI – typical images for DES-affected human pre-corneal tear film shown with recording time after blink

Procedures employed for TFSQ indices assessment

Main research consisted of an automated procedure. Firstly, the determination of one spectral and four fractal TFSQ descriptor time vectors (with pre-processing, background filtration, and spatial averaging) has been performed on the interferogram sequences. This has been followed by simple or bisegmental linear regression and statistical analysis with extraction of 10 global (in the temporal division) features of tear film kinetics, which include three derived after linear regression and seven after bisegmental linear (for details see section 7.1.2 and table 7.1). During next stage, the statistical tests for the abovementioned features as well as for the regression residuals (from which additional two features have been extracted, giving a total of twelve features per measurement) have been performed. Afterwards, the statistical distributions of these parameters amid sequences have been derived within each of the two groups of patients: healthy and DES-affected. This has been done using kernel density estimation (KDE) using Epanechnikov's kernel as it is the most efficient, i.e., optimal in terms of the lowest possible asymptotic mean integrated squared error (AMISE) [100]. Finally, receiver operating characteristic (ROC) curves and areas under the curves (AUCs) have been calculated in order to compare visually and quantitatively the classifiers constructed using different TFSQ descriptors and statistical parameters of kinetics time series.

All of the descriptors have been calculated over the twenty 128 px × 128 px subframes (the original images are 704 px × 576 px). If contrast of a subframe has been detected to be too low (i.e., frame has been too dark), the frame has been rejected during the pre-processing stage, after the initial vignette reduction and other detrending and filtering operations. Similarly, each of the remaining subframes is investigated in terms of its 0-th and 1-st spatial spectral maxima and the depth of the valley between them – if it is too shallow (the image is too blurred), such a subframe is also rejected. This rejection procedure is unified for each TFSQ descriptor to ensure the maximum likeness of the experiment conditions and is performed with respect to the raw input image, i.e., that has not been subject to pre-processing, to provide firm restrictiveness of the removal. Correspondingly, the averaging of the subframes' results to frame's result is done in identical way for all sequences – the calculated values are: median, mean, and standard deviation. For the further analysis, the median has been chosen as the most robust, but the other ones have also been stored (in conjunction with each descriptor's calculation time per individual frame) to allow further analysis.

THRESHOLDING IN INTERFREOMETRIC DATA PRE-PROCESSING Data pre-processing is an issue of particularly vital importance especially in case of LSI based interferograms analyzed by methods operating on binary images in conjunctions with edge detection. In such cases, the boundaries – or contours – of the interferometric fringes need to be determined accurately in possibly the largest part of the input image. Then, the binarisation is the critical point of the pre-processing stage, as it destroys much of the data contained in the image. The goal is to retain as much of the useful data as possible in the structure of the monochromatic output while allowing the superfluous or redundant information to be taken out with the removal of the grayscale quantization levels.

In the simplest case, the binarisation threshold may be constant for the whole image and arbitrarily set to some value, e.g., half of the maximum image intensity. Using an algorithm to optimize the threshold value is considered a better approach and to exploit *Nobuyuki Otsu's method* [101, 102] is a popular solution with a large spectrum of implementations [103]. It is based on the minimization of the intra-class intensity variance that in a binary case is equivalent to the maximization of the inter-class variance [101].

A more advanced approach involves *adaptive thresholding* – a form of thresholding that takes spatial variations in illumination into account [104]. Locally adaptive image threshold may be calculated using pixel-based *local first-order image statistic* [105]. It can be also fine-tuned by changing *sensitivity* of determining which pixels belong to foreground and which to background, e.g., sometimes classifying more pixels as foreground may be desired, even though it can misclassify part of the background pixels. Another parameter of this method is the *neighborhood size* used in computing local statistics, typically a positive odd integer. As a rule of thumb, a size of approximately one eight of the image size is chosen as first approximation for giving satisfactory results, e.g., in MATLAB'S built-in `adaptthresh` function it is computed by default for image matrix `I` as `2*floor(size(I)/16)+1)` [105]. *First order local statistics* are used to compute local threshold value for a given pixel, i.e., a thresholding level calculated in regard to the neighborhood of this pixel. One example of such statistics is the *local mean intensity*, also referred to as *Bradley's* method [104]. Also, the *local median* may be applied, though it is regarded as relatively slow in execution and thus it is recommended to use this operator with smaller neighborhood sizes [105]. Another commonly used statistic is the *Gaussian weighted mean in the neighborhood*. The most appropriate statistic may be chosen experimentally or theoretically, e.g., if the distribution of the image data is known and may be proven to fit well with one of known operators according to the crite-

rion of interest. Further improvement in the results yielded by adaptive thresholding may be achieved by experimentation with various neighborhood sized to chose the optimal one for the investigated class of images. Adaptive change of the neighborhood size is also possible, but due to its complexity in terms of introducing internal layer of adaptability and therefore addition of a number of extra parameters to be optimized, it is of scarce use as its benefits rarely outweigh the computational and methodological overhead introduced by this method. A simpler and much easier from practical point of view approach may be to involve a neighborhood that is anisotropic, i.e., based on non" one square operator matrix. In applications like LSI-based images it may be particularly useful, as the intrinsic form of the data is characterized by a high level of anisotropy. E.g., it may be better to obtain the threshold by averaging the image along the interferometric pattern rather than perpendicularly to it and across many fringes. In the latter case, the lightness variability introduced by the fringe pattern itself could overshadow the more subtle (or low-frequency) intensity changes of the background level. Nevertheless, such fine-tuned optimization of the input data is of secondary importance in the task of comparing different FDE algorithms, as the data should be cleared enough to be of suitable input form for all of them, but not necessarily perfectly optimized. This would be of interest when a chosen fractal-based method is accepted for practical deployment, e.g., and prepared for industrial application – but such initial data optimization is not specific for fractal-based analysis. Therefore, the pre-processing stage has been investigated in terms of optimization to achieve a procedure that removes most of the unwanted background variations in a typical case, yet does not strongly alter or modifies the images by excessive processing. Moreover, efforts have been made to apply the same pre--processing routine for all of the studied algorithms, so that the found differences would not have been introduced by other factors than the properties of the investigated methods.

Figure 7.4 presents the influence of the thresolding procedure on the binarisation results for illustrative LSI data. This directly applies to algorithms requiring binarisation, but not only to them, as it is described below. Comparison between fig. 7.4a and fig. 7.4c shows the advantage which adaptive thresholding has over Otsu's--based constant approach in terms of revealing fringe pattern in the regions of image that have uneven brightness level. This is further confirmed by comparison between fig. 7.4b and fig. 7.4d that shows increase in the total length of detected edges in favor of adaptive method. This can be quantitatively assessed by counting the non--zero elements of both binary edge images that is a simple estimation of total length

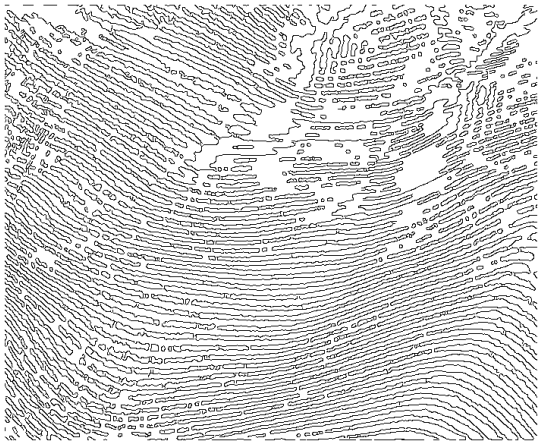
of the detected edges. Also, it may be noticed that as new edges have been detected when introducing adaptive thresholding, virtually no loss in detection of edges found with Otsu's approach may be noticed. Such analysis comparing the amount of detected edges and any potential losses for some regions has been the basis for optimization procedure that allowed for setting the actual pre-processing parameters used for further research. Particularly, reducing the neighborhood size about twice when compared to default value of MATLAB's adaptive thresholding, i.e., to about one sixteenth of the image size, has been found to further improve edge detection efficiency. This may be observed when further visual inspection of fig. 7.4 is done, i.e., comparisons between fig. 7.4c and fig. 7.4e and between fig. 7.4c and fig. 7.4f.

Figure 7.5 shows that, as it is mentioned above, the benefits of adaptive thresholding are available not only for methods requiring binary input. Instead of using the detected threshold as the decision level for binarisation, it may be used as reference value for background removal procedure. In some cased a subtraction of the adaptively estimated background level (with correction applied by, e.g., introducing a proper constant positive offset of resulting values) may mitigate the irregularities in the illumination of the scene. Comparison may be made by performing grayscale edge detection on the outputs and labeling the pixels in the following manner: first class for non-edges, second for the ones detected by both approaches, third for the ones detected only by the first approach and fourth for these detected only vy the second approach. Numbers of pixels within the second category may be counted to reveal the degree of the consistency between both results. The numbers of pixels within the third and fourth category may be counted to show their differential advantages. A visual insight may be accomplished by visualizing the results with a binary image of common edges on background with two contrasting colors imposed to mark edges detected only by single-approach. This allows evaluating of the results of the adaptive thresholding and background removal to be performed both quantitatively –basing on the values of the counted pixels – – and qualitatively – basing on visualizations similar to the one presented in fig. 7.5. Such analysis has been performed on LSI data to validate and refine the pre-processing parameters obtained during the binary-based analysis illustrated by fig. 7.4. When the edge detection is of particular interest, results of two or more approaches may be combined in a single map of edges – this could be suitable for pre-processing of binary methods like, e.g., edge binary box-counting (EBBC) approach, but not directly applicable for the grayscale ones. Therefore, such multi-approach pre-processing variants have not been employed in the described research.

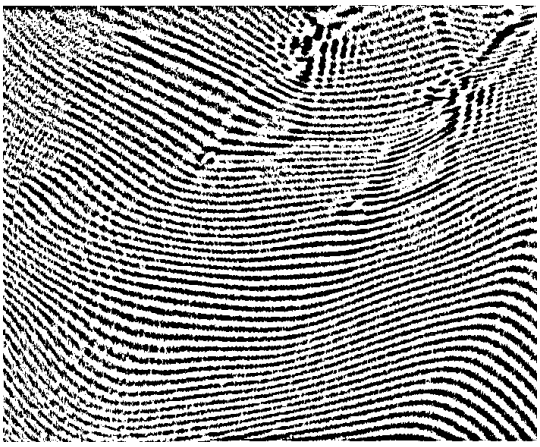
(a) Constant threshold calculated using Otsu's method

(b) Edges detected in (a)

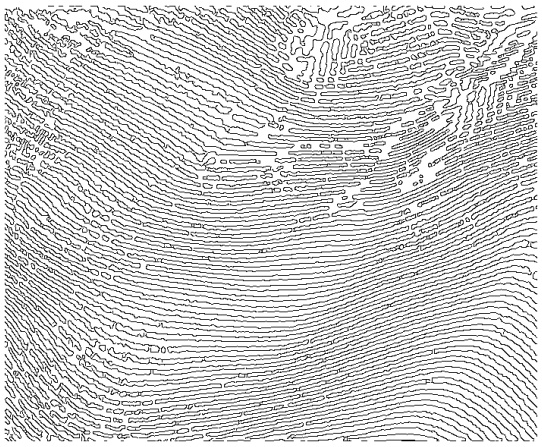
(c) Adaptive threshold calculated using MATLAB's `adaptthresh` function with default parameters



(d) Edges detected in (c)

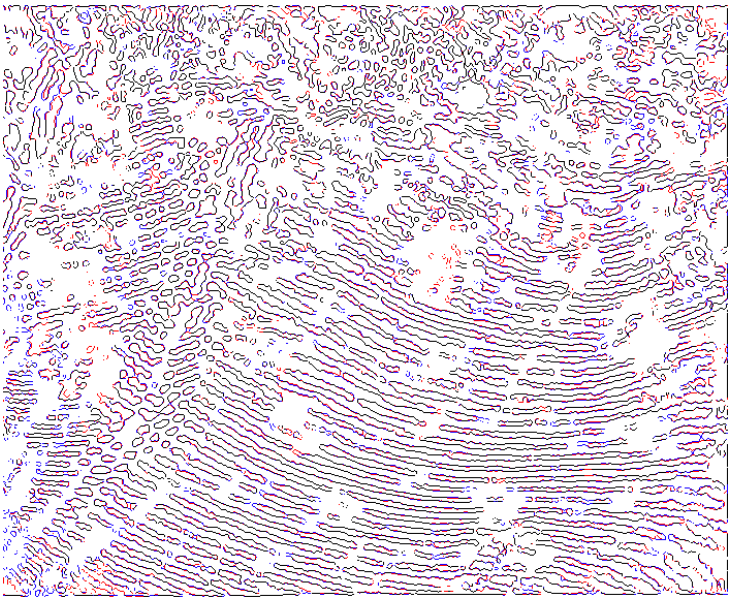


(e) Adaptive threshold with neighborhood about twice smaller than in (c)

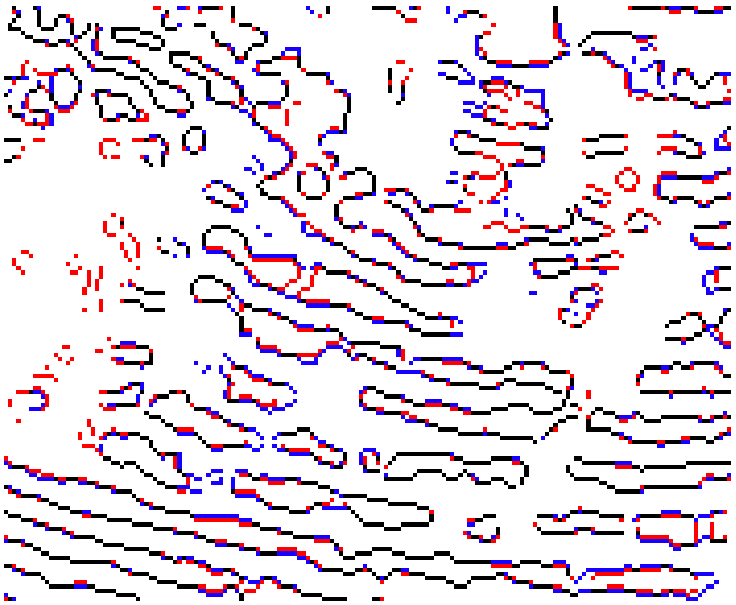


(f) Edges detected in (e)

Figure 7.4: Comparison of binarisation results for different thresholds and neighborhoods (for clarity of presentation, values of the gray pixels – including the black and white ones – are visualized in inverted intensity)



(a) Whole frame



(b) Enlargement of a part of the frame (one sixteenth of the original image by surface)

Figure 7.5: An illustrative comparison of edge detection in grayscale on LSI image – without vs with adaptive background removal (grayscale inversion has been applied as in fig. 7.4). Image data has been acquired for tear film surface of DES-affected patient. Pixels detected as belonging to edges in both images are rendered in black, while these detected only in original and only in pre-processed image – in blue and red, respectively

SPECTRAL DESCRIPTOR: $-M_2$ Descriptor $-M_2$ is additive inverse of spatial--average localized-weighted estimate of the first harmonic [94]. It belongs to a family of frequency-spectrum-based descriptors build upon the morphological analysis of the first-order carrier region of fast Fourier's transformation (FFT) of an interferogram. In the used form, the lesser the image periodicity (and thus the more deteriorated the tear film surface), the lower below zero would be the value of $-M_2$.

FRACTAL DIMENSION ESTIMATION For completeness and ease of readibility of this part of the thesis, **some theoretical aspects, described in detail in part II, are reiterated here**. The following fractal-based TFSQ measures have been incorporated into the research.

(1) FD_{BBC}: binary box-counting (BBC) – performed on the subframe image binarized using the adaptive threshold with optimized parameters. Optimization is similar to adaptive background removal during pre-processing and has been manually tuned for the best edge detection in a general case (giving possibly biggest total length of detected edges with acceptably low noise influence).

(2) FD_{EBBC}: EBBC – an application of box-counting (BC) method of FDE on image edges detected using Laplacian of Gaussian (LoG) method that – – for this class of images – has been identified to be more sensitive and robust than Sobel and less preferring closed loops than Canny operator. Edge detection is performed on an image binarized using the same method as for FD_{BBC} algorithm.

(3) $FD_{W\text{-}GBC}$: author's implementation and concept of weighted grayscale box--counting (W-GBC) algorithm utilizing information about the image before. It utilizes a specifically designed effective matrix algorithm for weighting box values during rescaling with halving the image size based on p-norms, q.v. eq. (7-1):

$$\|x\|_p \doteq \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}. \tag{7-1}$$

Namely, 1-norm, 2-norm and ∞-norm might be used using the preconfigured function parameters with the latter one being chosen as the default due to giving more accurate result despite lacking the advantage of efficiency. In practical implementations ∞-norm is based on the maximum operator, as in eq. (7-2):

$$\|x\|_\infty \doteq \max_i |x_i|. \tag{7-2}$$

The algorithm utilizes block processing to optimize computational demand of searching for the maxima, but nonetheless it is still over an order of magnitude slower than the matrix variant.

The presented results (apart from figs. 7.6 and 7.6A) have been gathered for slightly different (faster and less accurate) scaling method utilizing MATLAB's `imresize` function.

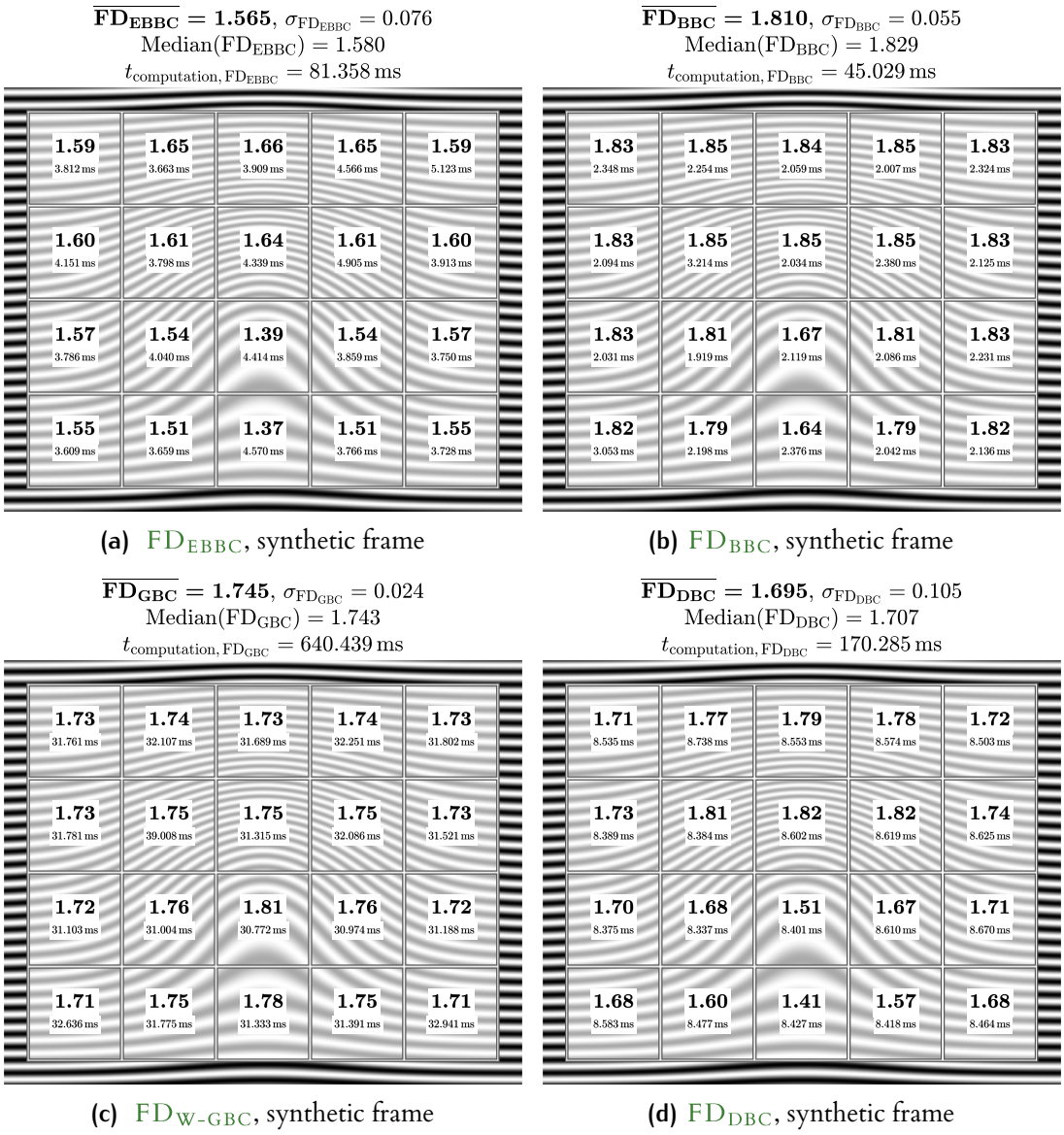
To maintain simplicity, when no other acgbc variants are considered than W--GBC, $FD_{W\text{-}GBC}$ is sometimes denoted as just FD_{GBC}. It applies to cases when W-GBC has been used as the sole representative of the family of grayscale BOX-COUNTING (GBC)-based methods, as in previous research its results have been assessed as the most promising. This consideration does not pertain to other grayscale-based methods whose abbreviations used in this thesis does not contain the 'GBC' core, e.g., differential BOX-COUNTING (DBC). For this purpose, these are classified as belonging to separate families of methods due to fundamental algorithmic differences between them and the narrow-sense GBC-based class.

(4) FD_{DBC}: author's implementation of N. Sarkar and B. B. Chaudhuri's DBC algorithm [17]. It uses information about image intensity as the third dimension (works in grayscale location-intensity space). This is the basic version of the algorithm without further numerical optimizations proposed by the original [31] and other [28] authors. That is probably one of the reasons of its poor empirical efficacy, as this class of methods might require specific randomization to maintain their sensitivity in presence of periodic patterns (frequent in LSI images) or small intensity variation.

Synthetic interferometric data analysis

SPACE-WISE ANALYSIS Thorough analysis of the output data of the synthesized fringe patterns has been performed in a mostly automated manner.

Figure 7.6 depicts the initial, but crucial stage of the image processing: subframe selection, TFSQ estimates calculation and averaging. Here, smaller numbers imposed below TFSQ estimator values describe empirically determined calculation times that are auxiliary information. A synthetically generated frame that includes centrally placed Gaussian phase disturbance is used as input data in frames (a) to (e). Spectral descriptor $-M_2$ is presented for frames (e) to (h) showing examples of different tear film classes (including simulated and real ones). There, contact lens frame (h) depicts tear film deterioration even worse than in case of DES-affected



(a) $\mathrm{FD_{EBBC}}$, synthetic frame

(b) $\mathrm{FD_{BBC}}$, synthetic frame

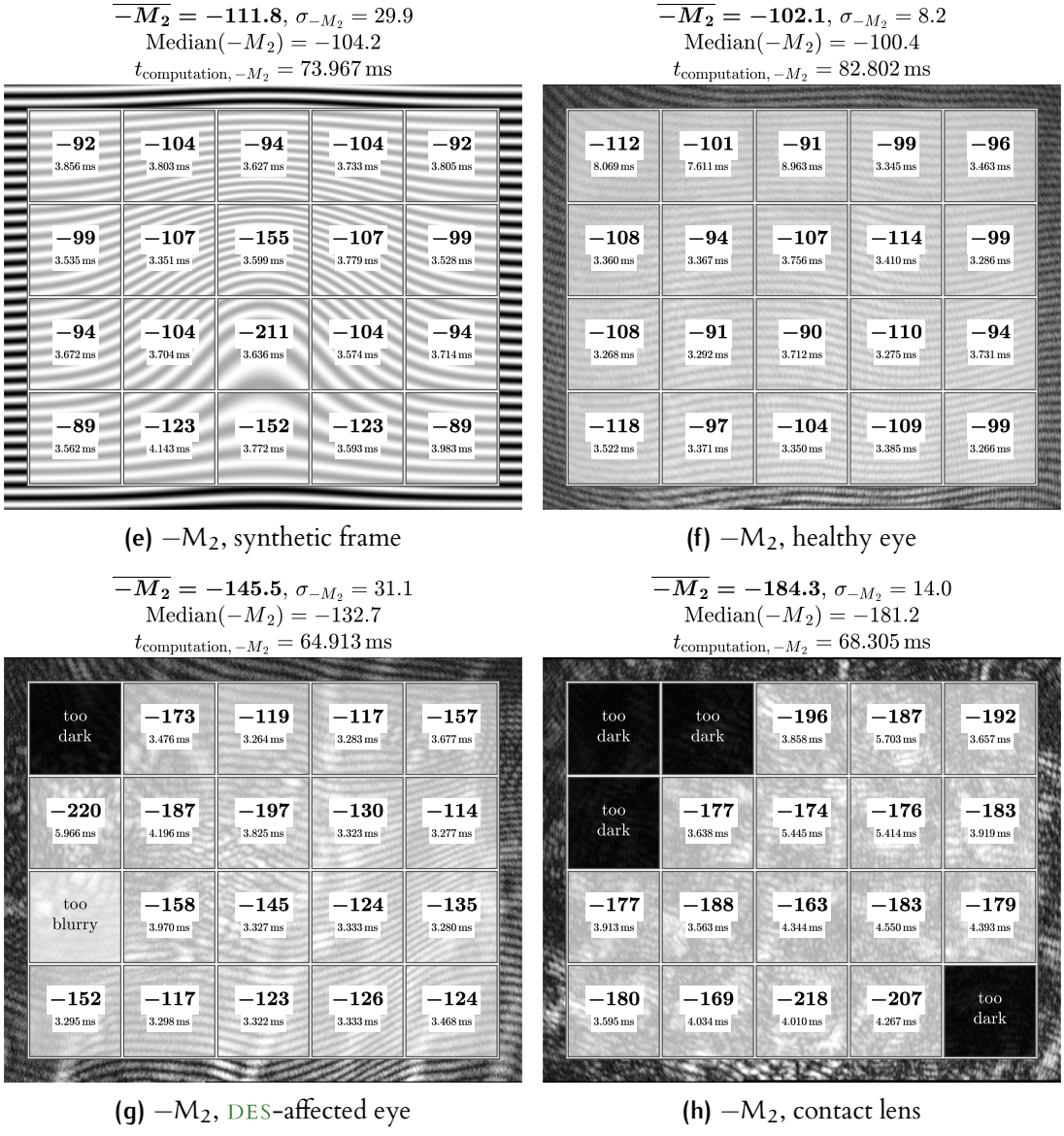
(c) $\mathrm{FD_{W\text{-}GBC}}$, synthetic frame

(d) $\mathrm{FD_{DBC}}$, synthetic frame

Figure 7.6: Interferogram frames visualization with subframe division and TFSQ estimates. For a synthetic frame, fractal-based descriptors (a) to (d) are used with spectral reference (e). Illustrative spectral descriptor values are presented for synthetic (e) as well as in-vivo frames (f) to (h)

eye (g) where, in turn, tear film is more degenerated than on the surface of a healthy cornea (f).

The obtained simulation-based results are presented in fig. 7.7. The purpose of this examination has been to assert that a basic fractal dimension (FD) estimator is able to measure slight frequency changes (that occur locally around tear film break-ups) and to determine the noise-like phase disturbance threshold that renders these measurements useless. Note that FD values for the fringes which have disruptions above the *noticeable* level are of low variability. Moreover, these val-



(e) $-M_2$, synthetic frame

(f) $-M_2$, healthy eye

(g) $-M_2$, DES-affected eye

(h) $-M_2$, contact lens

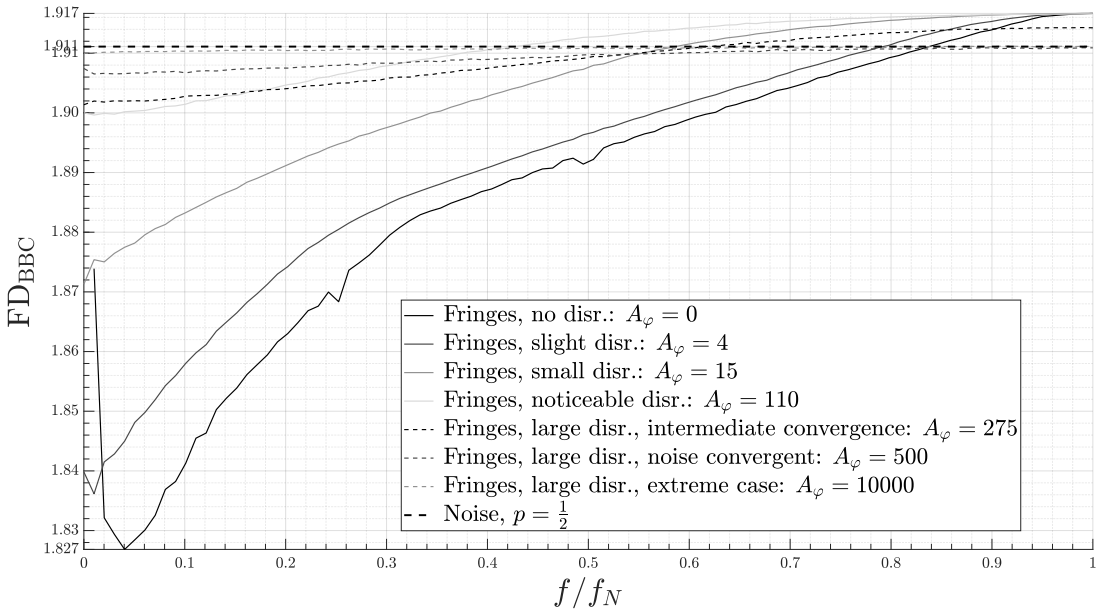
Figure 7.6A: Interferogram frames visualization with subframe division and TFSQ estimates. For a synthetic frame, fractal-based descriptors (a) to (d) are used with spectral reference (e). Illustrative spectral descriptor values are presented for synthetic (e) as well as in-vivo frames (f) to (h) (cont.)

ues converge to the noise level. This is especially visible in the enlarged part, i.e., in fig. 7.7b. Concluding, a reliable FDE-based TFSQ descriptor should react to changes in the structure of the tear film that are revealed by local fringe anomalies. Nonetheless, it ought not to overreact, i.e., it is supposed to interpret high disturbances (or even moderate ones occuring in the upper part of the frequency spectrum) as being indiscriminate against the random noise. As stated in section 6.1.1, the typical LSI of a tear film has dominating dominating fringes' frequency around one seventh of its Nyquist's threshold. Therefore, a major part of the image variability is expected to take place in the lower part of the frequency spectrum, at least until the interferometric image is not degenerated by extreme disturbances in examined tear film surface topography. These may introduce patterns that have much wider bandwidths and unpredictable peaks of their frequential components that cause the influence of the initial pattern frequency to be too faint to be plausibly detectable as also shown in fig. 7.7a.

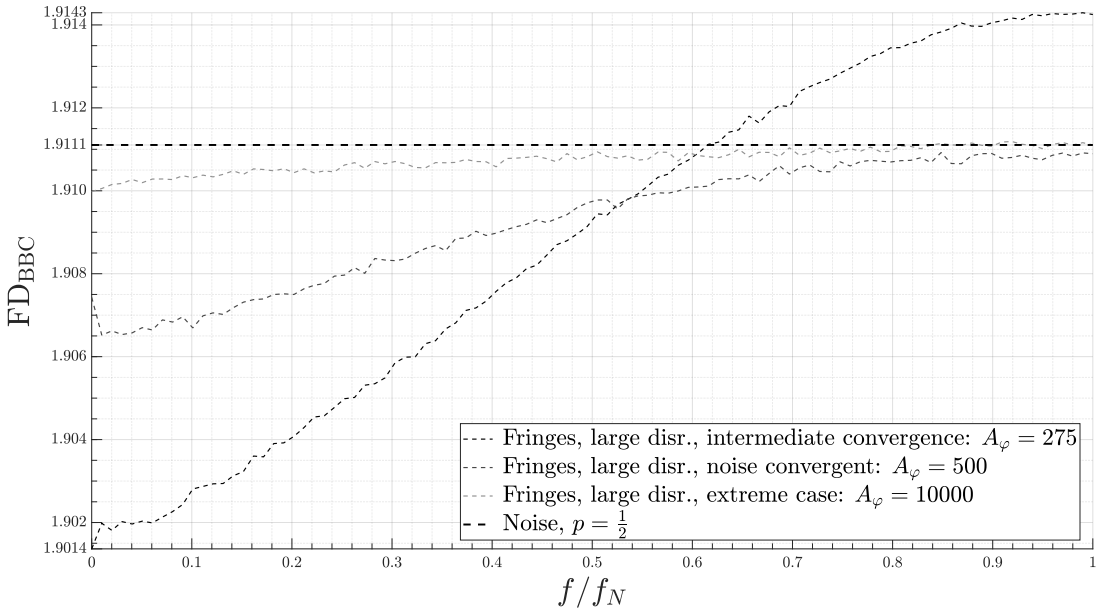
SUPPLEMENTARY SPATIAL-FREQUENCY-WISE RESEARCH Algorithm developed to visualize the dependency between spatial frequency of the interference fringes together with the amount of phase disruption in their periodic pattern returns a family of plots also storing the figure into a graphics file. In the produced picture one can distinguish two types of convergence (cf. fig. 7.7). They are interdependent with two classes of structural fringe disruptions described in the following text.

(1) The first class, within which the introduced pattern disturbance is of slight to moderate degree that do not prevent FD estimator of the image to converge to the value noticeably higher than a characteristic threshold with increasing frequency. This stabilization level has been determined empirically to be the expected FD estimator value for a binary noise with the probability of taking value of 1 with probability $p = \frac{1}{2}$. That makes this class of images distinguishable from noise even for very high frequency with a level of disruption that visually might appear to be close to quasi-random signal. This sub-group of plots includes:

   (1a) complete lack of disruption – an ideal case (sinusoidal phase disturbance amplitude equals to 0);
   (1b) noticeable disruption (sinusoidal phase disturbance amplitude equal to 4);
   (1c) moderate disruption (sinusoidal phase disturbance amplitude equal to 15);



(a) All investigated cases



(b) Enlargement of the ordinates (vertical) axis around the random noise $\mathrm{FD}_{\mathrm{BBC}}$ level asymptote

Figure 7.7: Supplementary simulative experiment results – $\mathrm{FD}_{\mathrm{BBC}}$ estimates for different fringe frequencies (from zero to Nyquist) at different disruption levels

(1d) significant disruption (sinusoidal phase disturbance amplitude equal to 110).

(2) The second class, for which the disorder grade is sufficiently high to make FD estimator of their pattern converge from below to the aforementioned noise level. Plotted values of vectors tend not to exceed this noise threshold by more than a relatively small (due to intense averaging) estimator variance margin. This class includes:

(2a) strong disruption (sinusoidal phase disturbance amplitude equal to 500);
(2b) extremely strong disruption (sinusoidal phase disturbance amplitude equal to 10 000).

In the axes there is also an additional plot for a big disruption of intermediate convergence not belonging completely to any of the two groups mentioned above. Its sinusoidal phase disturbance amplitude has been set to the value of 275 that places it in-between of these two, otherwise well-separated, classes.

Moreover, to increase the readability of the plot, the referential FD value corresponding to the random binary signal with 50 % probability of success in Bernoulli's trial of each data point has been indicated with thick dashed black line.

Real data analysis – software implementation

The input video file represents a time series and thus while internal TFSQ estimation algorithms calculate their output values, they operate on individual images and on not the whole sequence at once as in the fig. 7.8 depicting the data flow in numerical procedures. The first step of data analysis is reading of AVI video file containing LSI sequence. Then, detection and rejection of blinks as well as blur areas or whole frames (by discarding subframes of insufficient contrast) is performed. The next data processing stage involves the calculation of time series with TFSQ indices including four fractal-based for investigation and one spectral-based for reference. Following this, adaptive linear/bisegmental linear fitting is applied. Finally, statistical analysis providing 10 parameters regarding regression trends and the distributions of the residual vectors is performed.

Additionally, as blink periods are detected before the pre-processing stage, blink-classified frames are discarded as well as frames containing more than 6 subframes (out of 20 subframes in total) rejected due to blur. Thus, a set of images from AVI file to be further analyzed might be much incomplete and the total amount of time needed to process two apparently comparable videos may substantially vary.

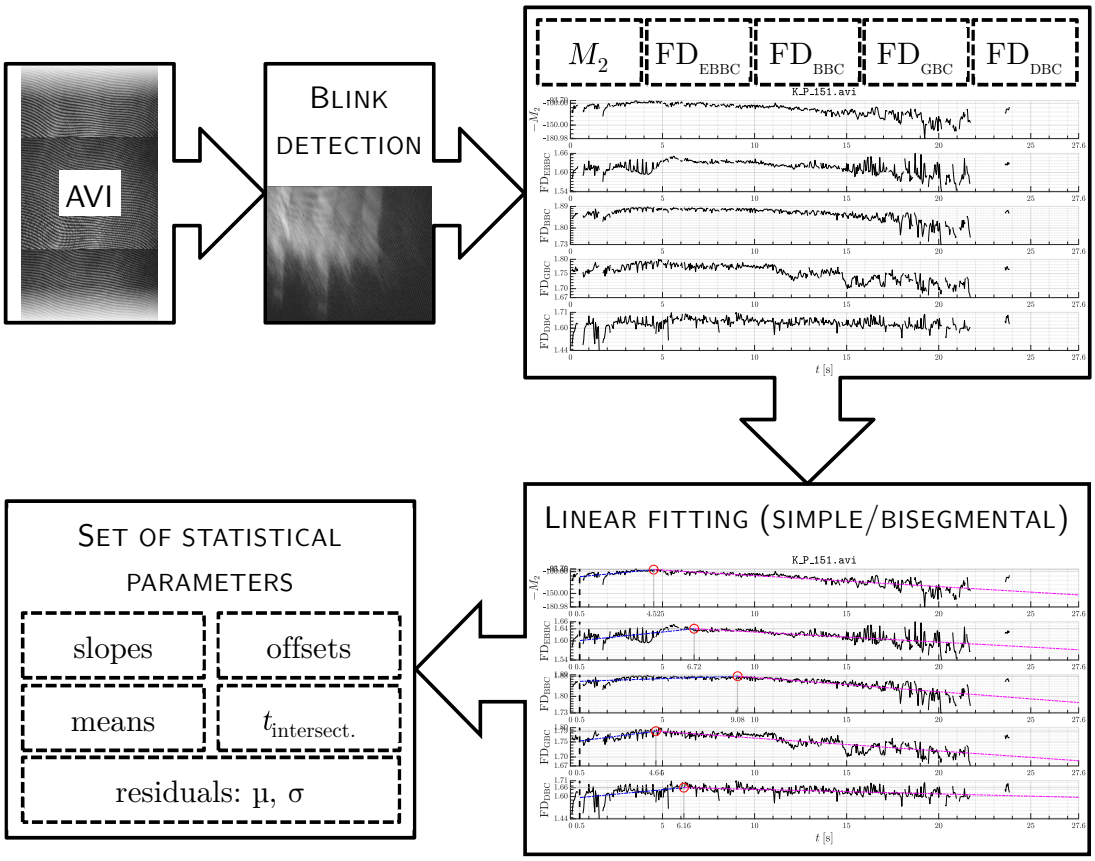


Figure 7.8: The automated data processing flowchart for the main algorithm. Major steps include: (1) reading AVI file, (2) blink and detection/rejection, (3) indices calculation, (4) adaptive simple/bisegmental linear fitting and (5) statistical analysis providing 10 parameters regarding regression trends and distributions of residues

Methods for statistical analysis

The battery of various statistical tests is used during the analysis process. Initially, log-likelihood-ratio test is performed to determine whether linear regression would remove significant amount of useful information regarding dynamic changes of TFSQ time series. Besides, the outliers' removal procedure by means of modified (iterative-recursive) Thompson's τ method is employed. After applying the suitable regression model, the descriptive statistics are calculated, with the supplemental role of one-sample Kolmogorov–Smirnov's test for normality (that might be replaced, e.g., by Jarque–Bera's test) to confirm the adequacy of the simple or bisegmental linear model. Table 7.1 presents includes the set of the 10 statistics calculated to find the descriptor used for classification along with two additional parameters derived from regression residues.

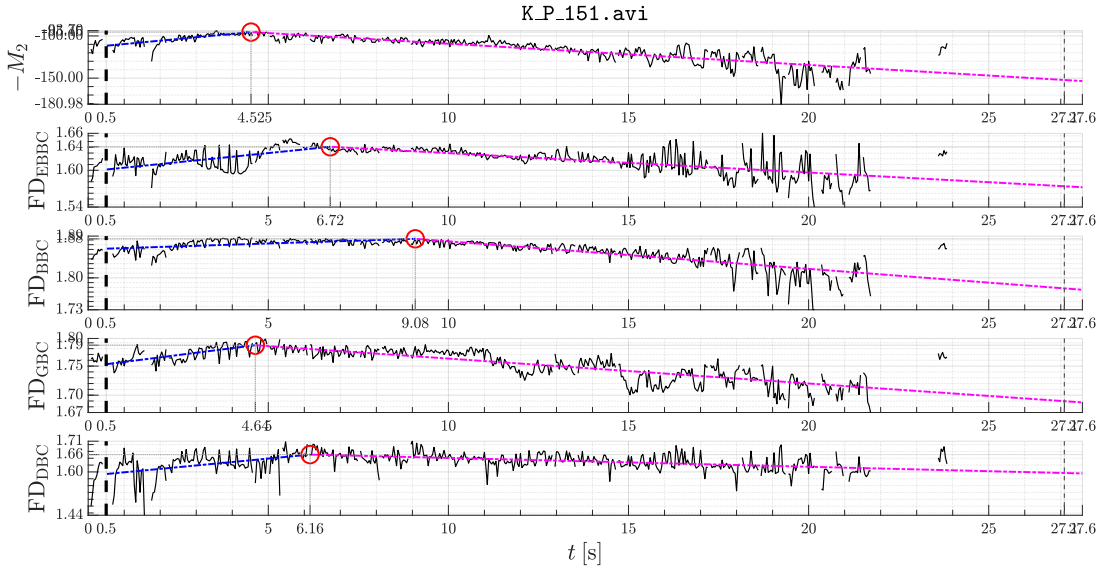
Computational aspects

Numerical computations have been performed in MathWorks MATLAB 9.0 R2016a up to 9.7 R2019b Prerelease environment on a 64-bit MS Windows 10 PC with the hardware configuration encompassing the following components: Intel Core i7-4910MQ CPU at 2.90 GHz, 32.0 GB RAM (dual-channel DDR3 SDRAM at 798 MHz) and SSD mass memory (1 TB Samsung 850 EVO).

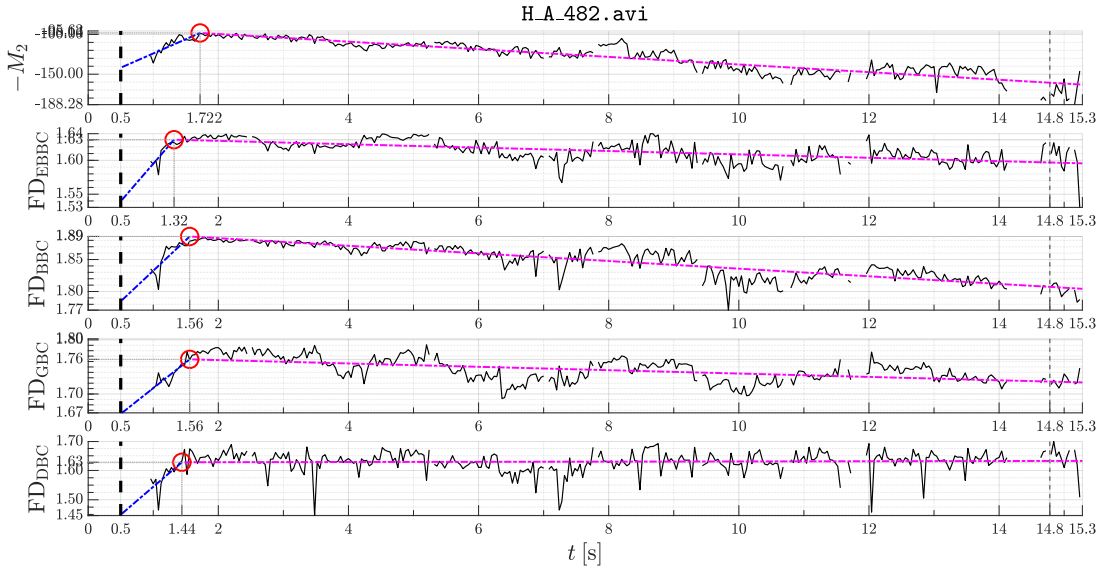
7.1.2 Results and analysis

The obtained TFSQ time vectors have been subject to intermediate processing, i.e., initial 0.5 s margin excluding it completely from regression (due to frequent and large instabilities of the tear film during early buildup) and final 0.5 s margin excluding it from bisegmental linear regression intersection point searching, but not from the regression itself (due to excessive risk of overfitting bisegmental linear trend to a few data points if the rest of the vector is flat enough). Two examples (healthy eye and DES) of visualization of this approach and its results are shown by fig. 7.9.

The final stage, after performing regressions, statistical tests, and KDE of the extracted TFSQ features, encompasses ROC curve and its AUC value calculation – – the closest the value would be to 100 %, the better the classifier build upon specific TFSQ descriptor and its time vector regression feature (50 % represents a random guess and 0 % – the anti-perfect classifier). The results can be presented in a way easy to graphically interpret and cases of some relatively plausible classifiers is depicted

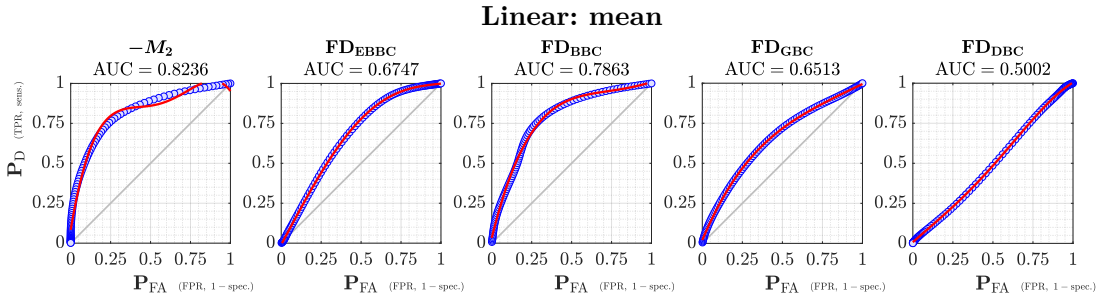


(a) Normal subjects

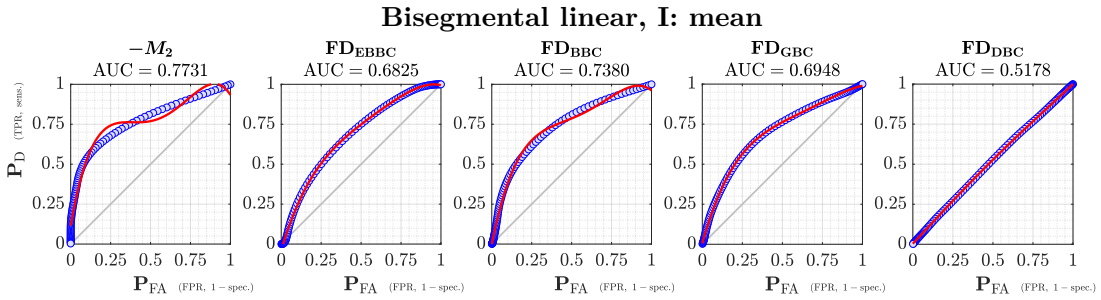


(b) DES subjects

Figure 7.9: TFSQ index time series estimation using different approaches – illustrative cases where spectral- and fractal-based procedures for descriptor estimation performed similarly well returning consistent results – *a typical situation*



(a) Best of all cases – mean value of linear regression: 82.4 %



(b) Best of bisegmental linear cases – mean value of the 1-st section of bisegmental linear regression: 77.3 %

Figure 7.10: Illustrative ROC curves – two of overall best cases with highest AUC value for the whole set of descriptors ($-M_2$ and all FDs)

in fig. 7.10. Plotted graphs represent classifiers constructed upon five different TFSQ estimators by using their mean value from the linear regression as a discriminant feature.

AUC percentages calculated for all 10 parameters derived after regression together with two parameters of residuals are shown in table 7.1. The best classifier in terms of AUC would be created of $-M_2$'s linear regression means (82.4 %), but the equivalent of FD_{BBC} (78.6 %) is probably not significantly worse, specifically if one takes into consideration its other ascendancies described below.

One-sample Kolmogorov–Smirnov's test (with the null hypothesis that the data comes from a standard normal distribution) has been used to assess the normality of the regression residuals. This has been motivated by the need for confirmation of the data homoscedasticity that is required to recognize the linear regression approach as an appropriate method of parameter estimation. The results are presented in table 7.2.

The information that the majority of TFSQ time vectors (for which the null hypotheses have not been rejected) might be treated as having the same kind of the distribution of the residual lead to a question whether parameters of their distribution, namely mean value and variance, carry any useful information that might

Table 7.1: AUC values of ROC curves for considered DES detectors: 10 statistics of TFSQ time distribution together with two parameters of the distribution of TFSQ residuals; 100 % corresponds to the ideal classifier, 0 % to random guess

Regression		Time vector parameter	Area under the curve / %					
Type	Segment		$-M_2$	FD_{EBBC}	FD_{BBC}	$FD_{W\text{-}GBC}$	FD_{DBC}	Average of FDs
simple linear	whole	slope	67.4	51.9	59.5	46.9	65.5	56.0
		offset	62.1	65.2	67.1	66.7	53.8	63.2
		mean	82.4	67.5	78.6	65.1	50.0	65.3
bisegmental linear	1^{st}	slope	54.6	42.8	54.1	58.7	46.9	50.6
		offset	68.8	53.8	55.7	52.9	46.5	52.2
		mean	77.3	68.2	73.8	69.5	51.8	65.8
	2^{nd}	slope	62.7	57.8	58.4	46.7	55.8	54.7
		offset	53.1	60.3	61.3	54.0	54.9	57.6
		mean	79.8	62.1	76.6	66.4	48.2	63.3
	whole	$t_{breakpoint}$	59.1	60.3	63.8	52.5	50.7	56.8
residuals*	whole	mean	57.4	52.8	51.2	53.0	59.6	54.2
		variance	75.1	73.3	70.5	72.9	52.9	67.4

* Derived from either simple or bisegmental linear fit (depending on the result of log-likelihood-ratio test).

Table 7.2: Fractions of one-sample Kolmogorov–Smirnov's descriptor distribution normality tests passed ($p < 0.05$) after outliers filtration within two groups of patients: control (69 subjects / 345 tests) and positively diagnosed for DES by experts (41 subjects / 205 tests)

TFSQ descriptor	Fraction of passed tests / %	
	CONTROL	DES
$-M_2$	91.3	87.8
FD_{EBBC}	92.8	97.6
FD_{BBC}	52.2	87.8
$FD_{W\text{-}GBC}$	94.2	92.7
FD_{DBC}	72.5	82.9
average	80.6	89.8

be exploited to compare or classify patient groups in terms of different TFSQ descriptors. To investigate this matter further, these two residual-based properties have been added to test set of statistical parameters.

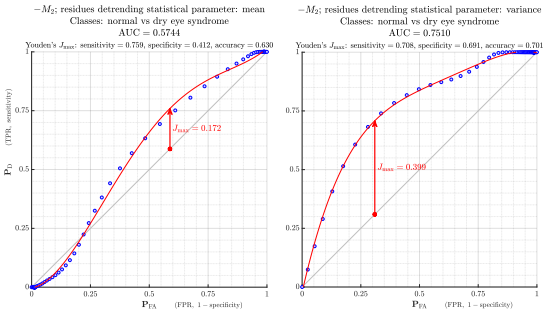
Before applying the normality test, the data has been filtered out of outliers. This removal has been performed using the modified (iterative-recursive) Thompson's τ method. Out of 550 measurements conducted in total, the worst case has been 18.1 % of outliers, and average has been 4.5 % with the best cases containing no detected outliers.

Wilcoxon's rank sum (Mann–Whitney's U) non-parametric test has been used due to relatively small sample size, q.v. table 7.3. Little sample constitutes a drawback in case of normality test that is required before applying a stronger parametric Student's t-test (as this test assumes of the distribution). Nevertheless, t-test has been performed to endorse the Wilcoxson's results, as they have been fully accordant, i.e., both tests have exhibited null hypothesis rejection occurring for exactly the same instances. For almost all instances of TFSQ descriptors the variance of residuals would serve as discriminant between healthy eye and DES, albeit in case of DBC algorithm the better choice would be sequence TFSQ mean value of the residuals.

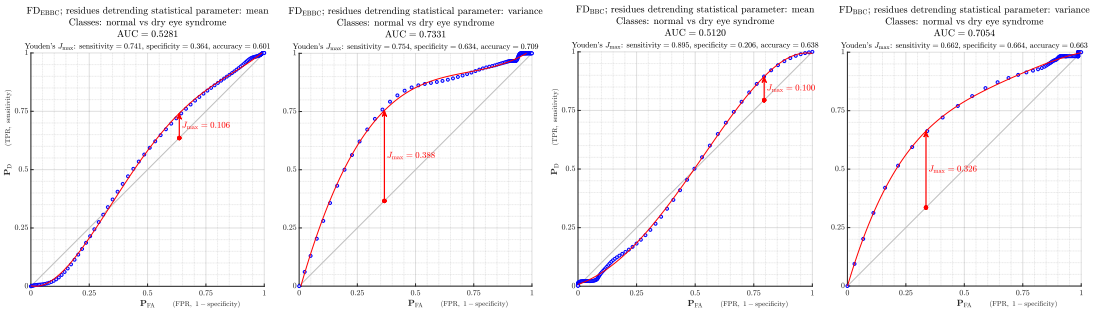
To perform the assessment of the diagnostic ability of a binary classifier system constructed upon the discussed parameters of residuals variation, a set of ROC curves has been constructed for them. For each characteristic curve in this set, AUC (the concordance statistic) and Youden's J statistic (informedness) have been determined. For more information about these summary statistics refer to appendix B. ROC curves together with the calculated statistics are presented in fig. 7.11.

Table 7.3: Wilcoxon's rank sum tests p-values for mean and variance values of distributions of simple or bisegmental linear regression residues. Highlighted (**bold**) are tests passed with respect to $p < 0.05$ criterion (i.e., whose null hypothesis about equal distribution medians for $\widehat{\mu_{res.}}$ or $\widehat{\sigma^2_{res.}}$ within both subject groups has been rejected)

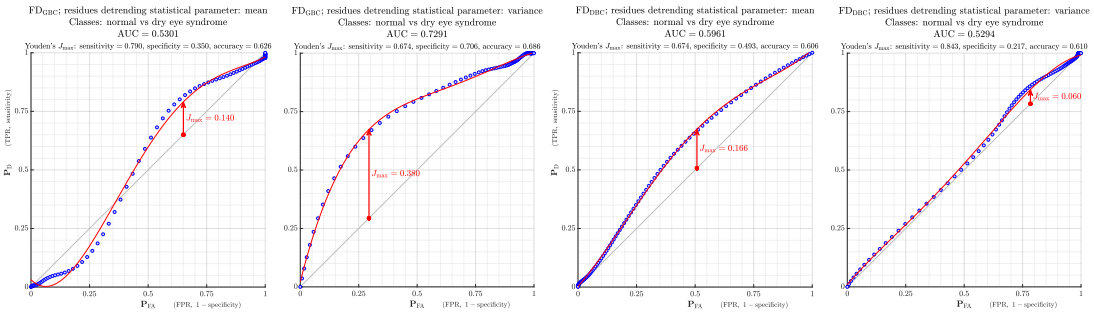
TFSQ descriptor	p-values for distribution parameters	
	$\widehat{\mu_{res.}}$	$\widehat{\sigma^2_{res.}}$
$-M_2$	0.143	**< 0.001**
FD_{EBBC}	0.447	**< 0.001**
FD_{BBC}	0.422	**< 0.001**
$FD_{W\text{-}GBC}$	0.711	**< 0.001**
FD_{DBC}	**0.029**	0.422



(a) $-M_2$



(b) FD_{EBBC} (c) FD_{BBC}



(d) $FD_{W\text{-}GBC}$ (e) FD_{DBC}

Figure 7.11: ROC curves for classifiers based on statistical parameters of simple or bisegmental linear regression residues of various BC-based FDE algorithms

Table 7.4: Computational efficiency analysis – statistical parameters of empirically obtained global execution times per LSI video sequence: the mean and the coefficient of variation estimates. Highlighted are the procedures evaluated as the fastest (**bold**) and the slowest (*italicized*)

TFSQ descriptor	Statistical parameter $\hat{\mu}$ / s	$\widehat{c_V}$ / %
$-M_2$	38.3	50.4
FD_{EBBC}	44.6	50.0
FD_{BBC}	**28.0**	49.8
$FD_{W\text{-}GBC}$	43.6	50.3
FD_{DBC}	*123.7*	50.4

Computational efficiency

LATERAL SHEARING INTERFEROMETRY DATA The last and additional stage of the main research has been to assess empirical computational demand of implemented algorithms. Results of CPU time measurements are shown in table 7.4 which presents averaged calculation times per AVI sequence of lateral shearing interferograms for five algorithms as well as variability coefficients of these values within algorithm classes. BBC algorithm proved to be the simplest, in terms of implementation, and thus the fastest one among the considered set of algorithms. The results of other estimates, including the spectral one ($-M_2$, q.v. section 7.1.1), are similar, with the exception of DBC which computations have been substantially slower.

SYNTHETIC FRACTALS Table 7.5 depicts CPU computation times measured for three classes of synthetic fractal images generated using Dietmar Saupe's rescale-and-add (RAA) method (as depicted in fig. 5.1) of size 512×512 px. Time measurements have been averaged over D values ranging from one to two in steps of 0.05, i.e., the same D vector as used in figs. 5.2 and 5.3.

Figure 7.12 presents CPU computation times measured for RAA-based fractal images as well as for Gaussian noise images. Depicted in semi-logarithmic scale are the lowest computation times among a number of iterations equal to 1000 for images of side length L equal to 32 px, 316 iterations for L of 64 px, and 100 for all images of L larger than 64 px. In the right-hand-side end of the figure, about $k = 15.6$ to 15.7, it may be noticed that W-GBC in its basic variant WEIGHTED GRAYSCALE BOX-COUNTING based on max operator (W-GBC-MAX) overruns computational speed of DBC. Worth noting is that DBC algorithm

Table 7.5: Averaged (over D) computation times measured for three classes of Saupe's RAA--based synthetic fractals – square images of side length L equal to 512 px – with different values of lacunarity parameter r. Calculations performed in MATLAB environment

Lacunarity parameter r	Mean computation times / s				
	BBC	P-GBC-IMG	P-GBC-INT	W-GBC	DBC
$\sqrt{2}$	1.76	3.63	25.98	1.85	0.17
2	2.14	4.35	32.39	2.18	0.20
4	2.15	4.31	32.43	2.18	0.18
average	2.02	4.09	30.27	2.07	0.18

is referenced to in the literature as 'efficient' and 'fast' [17]. W-GBC-MAX utilizes max operator which is a special case of generalized mean with infinite exponent $p = +\infty$. Such special case has been implemented in MATLAB environment using `blockproc` function, and other cases with finite exponents can be implemented in a more efficient way using two multiplications of image matrix by an auxiliary matrix. E.g., if input image is $\mathbf{I}$ there may be defined a matrix $\mathbf{A}$ such that $\mathbf{AIA}^{\mathrm{T}}$ gives the desired results. The actual implementation of this technique is presented in listing C.1 (line 349). What is more, generalized means with infinite exponents, i.e., min and max operators, may be approximated with this approach by using arbitrarily large, yet still finite, substitutes. An example of such efficient implementation being significantly faster than both DBC and W-GBC-MAX especially for smaller images is shown in the figure as WEIGHTED GRAYSCALE BOX-COUNTING based on avg operator (W-GBC-AVG). It utilizes a case of generalized mean with exponent $p = 1$, i.e., arithmetic mean, but as long as the exponent is finite, it may be set to virtually any value without a significant impact on computational overhead of the procedure. Therefore, a fine approximation of W-GBC-MAX may be implemented using this improved efficiency approach to achieve computation times about two orders of magnitude faster than DBC for image side lengths L up to at least 2048 px.

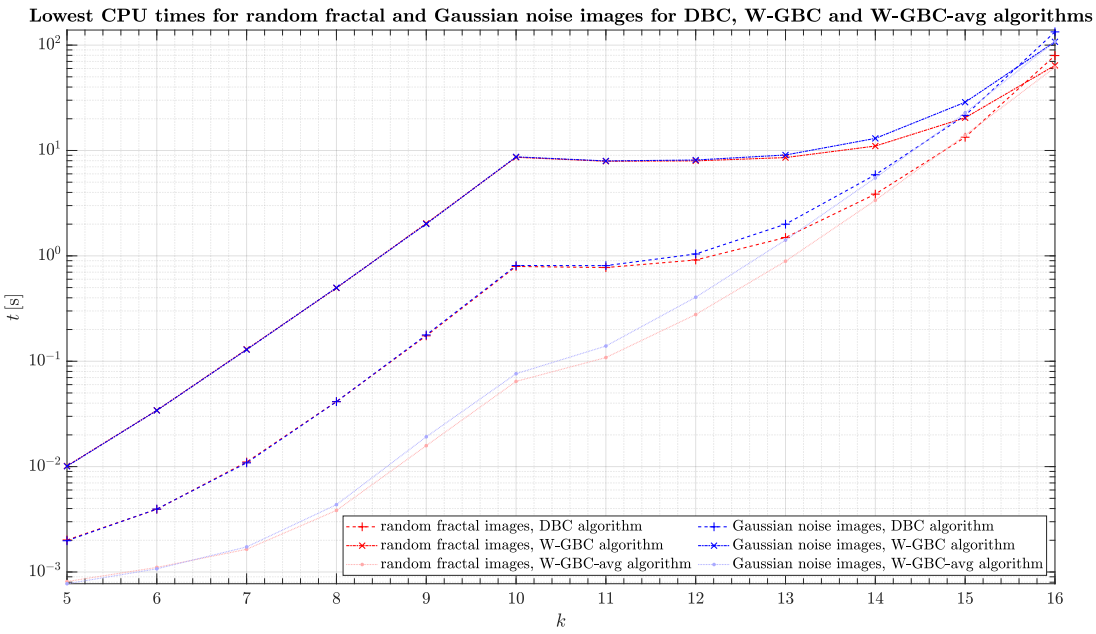


Figure 7.12: Calculation times for Saupe's RAA-based fractal images together with Gaussian noise images – minimal computation times among a number of iterations equal to 1000 for image side length L equal to 32 px, 316 iterations for 64 px, and 100 iterations afterwards (ordinates in logarithmic scale)

7.1.3 Coherence analysis

Additional implementation of coherence analysis procedure – frequential similarity

A preliminary research has been performed to demonstrate that coherence analysis constitutes a suitable tool to assess the resemblance between two time series of TFSQ descriptors in terms of frequency spectrum components. To determine whether their vectors undergo variations in a correlated manner and evaluate quantitatively the degree to which this similarity holds, one can exploit *coherence analysis*.

The algorithm utilized for generation of the examples presented in fig. 7.13 firstly interpolates any signal to be analyzed. This is done by applying piecewise cubic Hermite interpolating polynomials that complement the data lacking some elements or intervals of TFSQ vectors, as they have been generated only for frames that have met all the criteria described in 7.1.1. Specifically, motion-blurred frames have been selected and rejected; also, after frame segmentation, low contrast subframes have been dismissed. Then, linear detrending is performed on both signals to be compared to remove offsets and constant first order variability. Thus, cyclic features are exposed. Subsequently, *magnitude-squared spectral coherence* coefficient

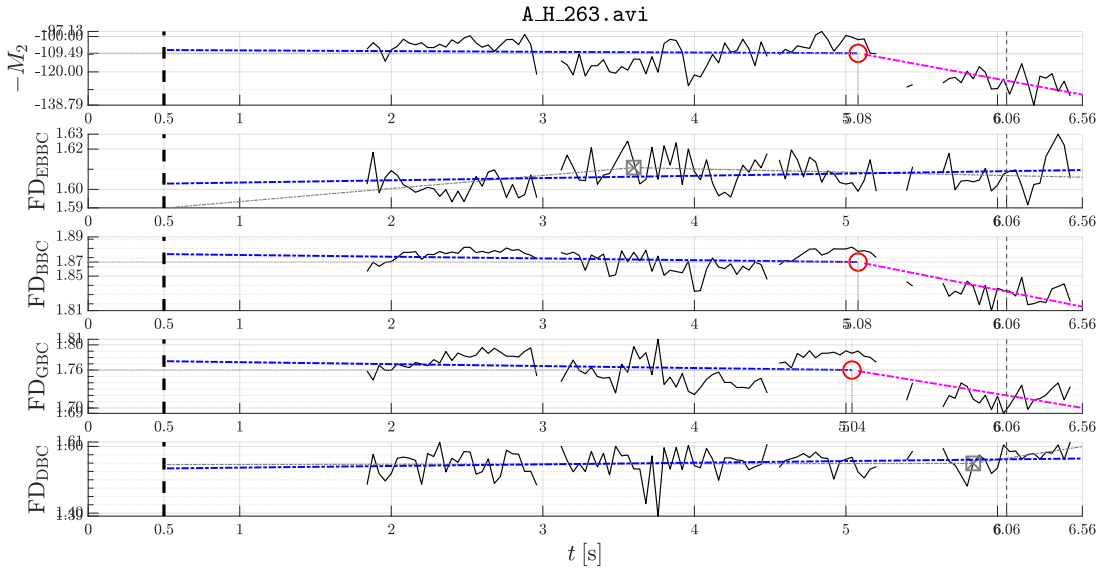


Figure 7.13: TFSQ index time series estimation using different approaches – an illustrative case of a normal (i.e., DES-free) subject where spectral-based procedure for descriptor estimation performed better than its fractal-based counterparts. Plot data of $-M_2$, FD_{EBBC} and FD_{BBC} has been used as input for analysis depicted in fig. 7.14

function $C_{xy}(f)$ is estimated numerically by using MATLAB's `mscohere` tool. During that process, Philip D. Welch's averaged modified periodogram method [106] is employed with signal being split into eight segments with 50 % overlapping and Hamming window of one-segment length. Finally, the script plots dependency between spectral coherence and signal frequency as a figure. One reference signal is to be chosen in advance (namely, usually it would be M_2-based spectral descriptor used as a benchmark for fractal-based ones) and two other TFSQ measures to be compared with it. Both signals are compared with the referential one. Two coherence plots are graphed into the figure to show how these two signals differ from the reference in respect of their frequential spectra. A sample graph of *spectral coherence analysis* is provided in fig. 7.14. As the degree of similarity between signals, as perceived by the observer, is only a rough approximation of the real amount of information that both of them share, such coherence is often used as an objective measure of the correspondence between time series. The illustrative coherence analysis in fig. 7.14 encompasses two pairs of TFSQ estimate vectors from fig. 7.13: $-M_2(t)$ with $FD_{EBBC}(t)$ and $-M_2(t)$ with $FD_{BBC}(t)$. It supports the conclusion that the later pair is more closely connected. This, in turn, is consistent with an observation that in a typical case the more visually similar two signals are, the more coherent they tend to be (within at least a significant part of their spectra).

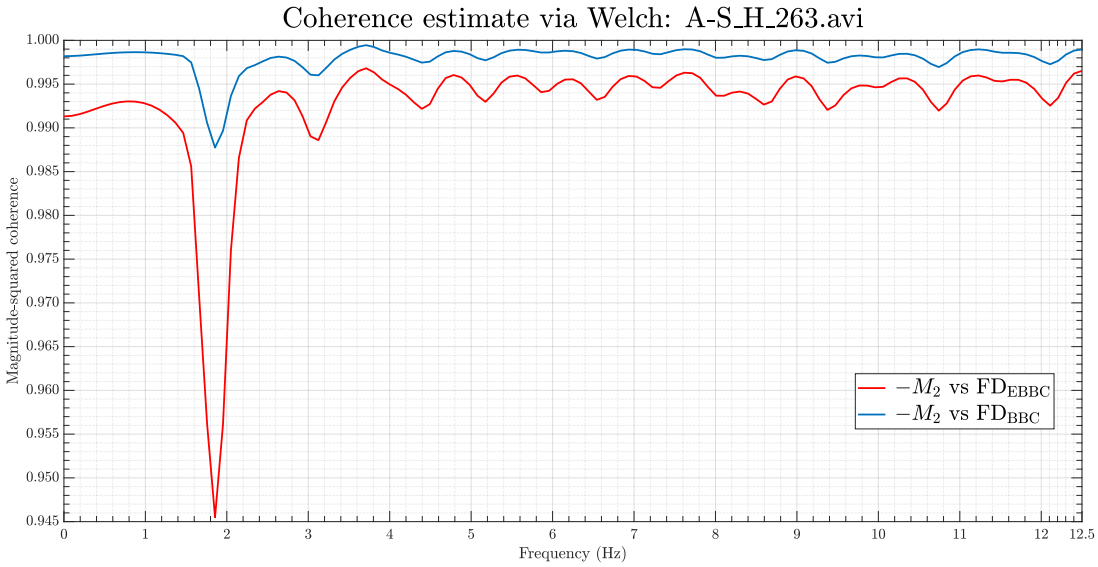


Figure 7.14: An example of coherence analysis plot for $-M_2$ with FD_{EBBC} and FD_{BBC} (based on data from fig. 7.14)

Partial conclusions and future recommendations

The conducted research together with the analysis of the results have led to three main conclusions:

(1) the application of FDE to interferometric image analysis is relevant;
(2) the differences present in FD definitions [1, 6, 7] has significant impact on differentiative abilities of and classification methods utilizing FDE algorithms based on different definitions or different estimators of the same quantity;
(3) there is at least one empirical FD-based TFSQ parameter (FD_{BBC}) providing classification power comparable to the reference frequency-spectrum-based algorithm (, q.v. section 7.1.1) whereas maintaining at least two aspects of prevalence over its frequency-spectrum-based counterpart:
   (3a) shorter time of calculations that indicates better computational efficiency (as in case of FD_{BBC}, nonetheless this does not hold true for other examined algorithms, q.v. table 7.4);
   (3b) possibly much stronger robustness to aperiodicity and thus disruption-proofness caused by the independence of FD-based TFSQ estimation of the possibility and accuracy of the spatial carrier frequency determination, whereas this constraint strongly limits the applicability area of the spectral algorithm class.

It is unlikely to experimentally explore all available FDs by systematically investigating their estimators. Nonetheless, a set of the ones chosen during the following

study has been proposed. There might also exists a new BC FD estimator well--suited for this particular application that could be designed basing on the theoretical analysis as well as numerical experiments.

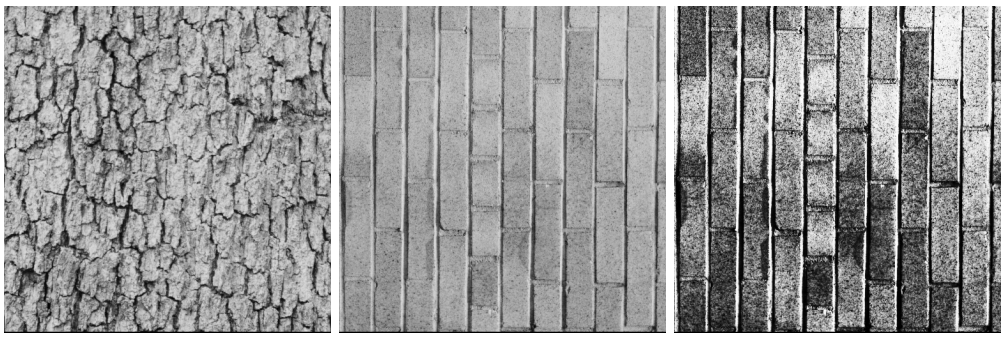
FUTURE RECOMMENDATIONS The future work may encompass:

(1) to propose *set criteria* which could be employed for the specific image processing application as currently there are no such widely-established standards known;
(2) to define clear conditions for deciding which FD estimate is best in terms of some image processing aspects; this possibly would be difficult to accomplish due to multicriterial estimator choice with possibly mutually exclusive optimization scenario, e.g., when minimum-variance unbiased estimator (MVUE) would not exist:
    (2a) Alternatively, adopt a simpler yet perhaps promising approach of extensive experimental comparison of carefully selected FD estimates and their utility in desired applications.

7.2 USC-SIPI IMAGERY*

Figure 7.15 depicts six illustrative USC-SIPI test textures. USC-SIPI database contains various textural images together with high altitude aerial photographies, miscellaneous generic imagery, e.g., animal and vehicle pictures, as well as some short image sequences. Of them, images containing textures has been of interes for the described research. The said database has been developed as a collection of digitized images at the Signal and Image Processing Institute of the University of Southern California. Since its initial distribution in 1977 it has been expanded a few times and, as it has been shared for usage in research purposes for many decades, now it is a popular test set serving for experimental purposes. Due to the described circumstances, this database constitutes an useful texture data source for *benchmarks* involving analysis of the image processing algorithms. Prior to the usage of USC-SIPI textures collection as an input to FDE algorithms, 25 texture images of size 1024 px × 1024 px from the database have been downscaled to match 512 px × 512 px size of the remaining 39 texture images. Subsequently, all 64 texture images of size 512 px × 512 px have been con-

* This section is partially based on author's coäuthored paper submitted to *IEEE Access* [62].



(a) Brodatz - Bark (D12)

(b) Brodatz - Brick wall (D94)

(c) Brodatz - Brick wall (D94 H.E.) histogram equalized version of (b)

(d) Tile roof

(e) USCtexture mosaic #2 (info) gray-level texture map with regions of four different sizes

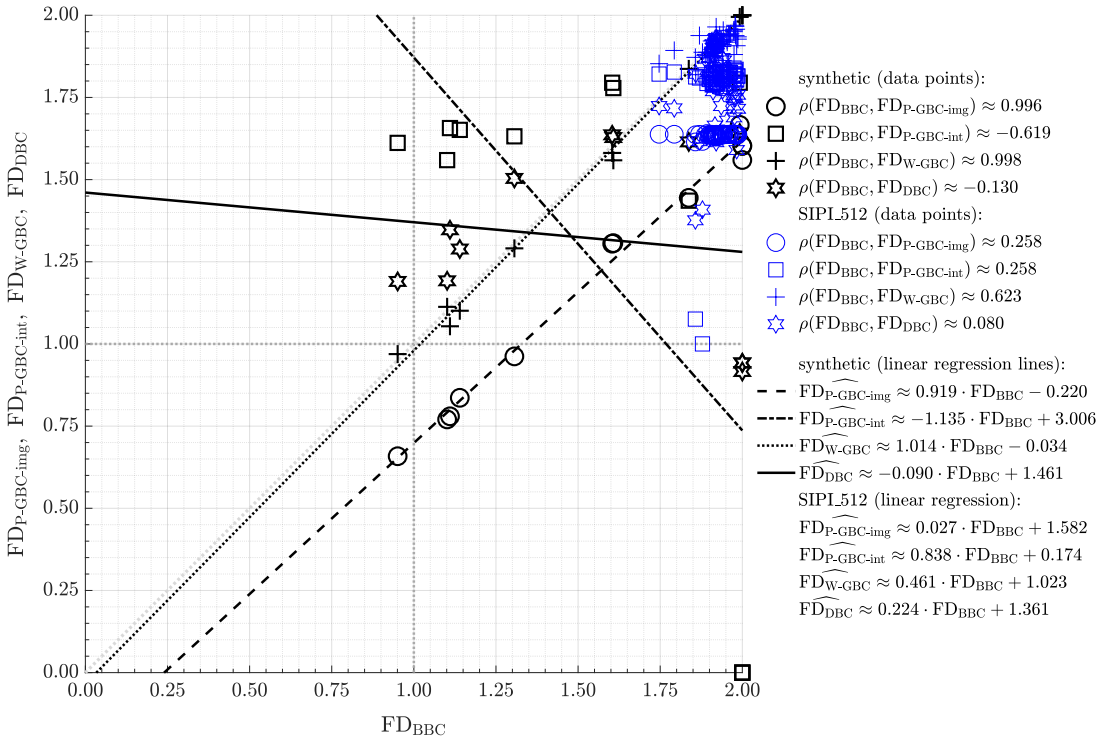
(f) USC texture mosaic #3 (info) gray-level texture map with non-horizontal and non-vertical texture boundaries

Figure 7.15: Illustrative images from the set of 64 textures from USC-SIPI database of 155 test images in 8-bit grayscale [107]

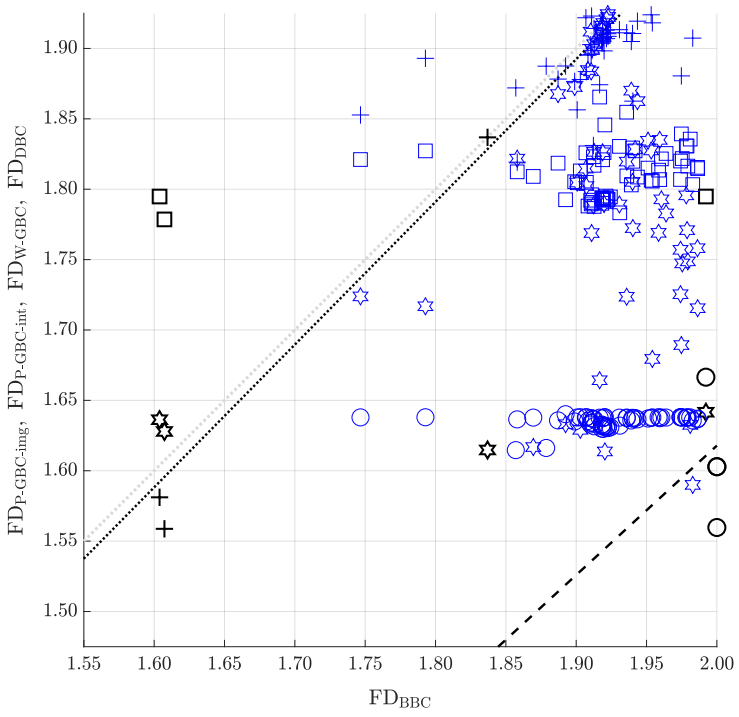
verted to grayscale and saved as image files of lossless portable network graphics (PNG) format to be read by image MATLAB environment and further processed.

Figure 7.16 depicts a comparison of results of FDE performed using various algorithms: probabilistic GRAYSCALE BOX-COUNTING based on image probability space (P-GBC-IMG), probabilistic GRAYSCALE BOX-COUNTING based on intensity probability space (P-GBC-INT), W-GBC, DBC and BBC. Estimation has been done for the set of 12 *simple* synthetic images, q.v. fig. 3.4 in section 3.2.4, that are not to be confused with synthetic *fractal* images that are furhter elaborated on in section 7.1.2, cf. fig. 5.1, as well as for USC-SIPI 8-bit grayscale test textural image data [107].

A distinction may be made between two approaches to FRACTAL DIMENSION ESTIMATION that exploit $FD_{P\text{-}GBC\text{-}IMG}$ descriptor to distinguish them clearly.



(a) Synthetic images and USC-SIPI textures



(b) Synthetic images and USC-SIPI textures – enlargement of the clusters of USC-SIPI data FD estimates

Figure 7.16: Comparison of FD estimates calculated using different methods: $FD_{P\text{-}GBC\text{-}IMG}$, $FD_{P\text{-}GBC\text{-}INT}$, $FD_{W\text{-}GBC}$ and FD_{DBC} with FD_{BBC} for 12 synthetic images presented in fig. 3.4 and 64 USC-SIPI textures (8-bit grayscale test images) [107] presented with examples in fig. 7.15

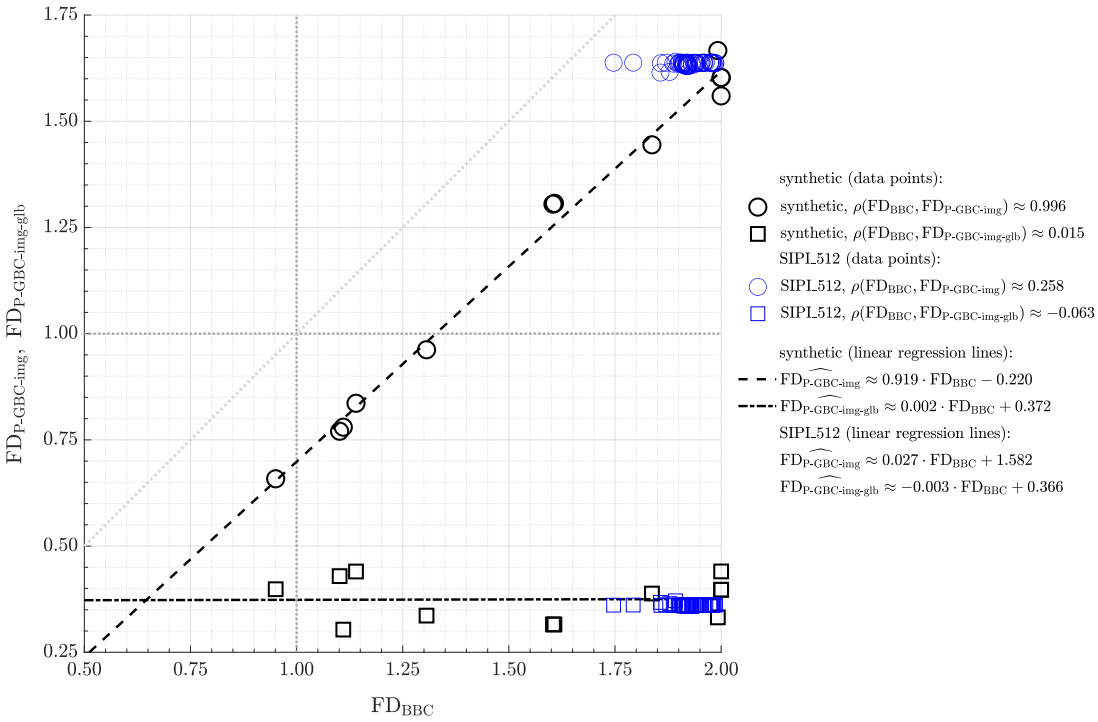


Figure 7.17: Comparison of FD estimates calculated using $FD_{P\text{-}GBC\text{-}IMG}$ and $FD_{P\text{-}GBC\text{-}IMG\text{-}GLB}$ estimators for 12 synthetic images presented in fig. 3.4 and 64 USC-SIPI textures [107] presented with examples in fig. 7.15

Namely, such discrimination can be useful to discriminate P-GBC-IMG and probabilistic GRAYSCALE BOX-COUNTING based on image probability space: globally calculated variant (P-GBC-IMG-GLB)† methods. Figure 7.17 depicts the result of comparison between them for synthetic and USC-SIPI data. It should be noted that FDE algorithm performance might be strongly dependent on the kind of the input imagery data, as P-GBC-IMG variant has been rather consistent with the classical BBC approach for the synthetic data while producing result being practically uncorrelated with the output of the referential BBC algorithm for USC-SIPI textures. Conversely, P-GBC-IMG-GLB variant of the method has not performed well for both classes of images.

In conclusion, it must be emphasized, that *the class of the images being under investigation is of substantial importance for the course of FDE and its results*. Nevertheless, when comparing figs. 3.5, 3.7, 5.2, 5.3, 7.16 and 7.17, a finding may be also drawn that regards only the classes of images that do not comprise data of purely artificial origin. Such restriction excludes both the simple and Saupe's synthetic classes.

† For the definition of P-GBC-IMG-GLB – as well as probabilistic GRAYSCALE BOX-COUNTING based on intensity probability space: globally calculated variant (P-GBC-INT-GLB) – see chapter 4.

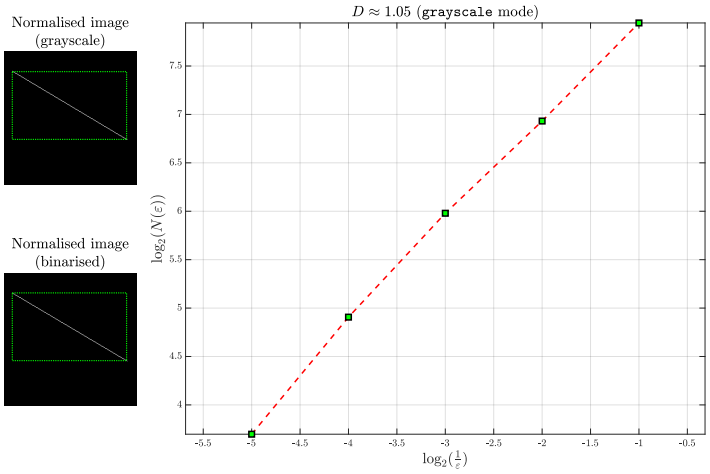
They are illustrated by fig. 3.4 and fig. 5.1, respectively. That is to say, for such subset of input data not all dependencies might rely on the peculiar structure of the images. In particular, vast uncorrelated regions of clusters containing data points for DBC and – at least to some degree – P-GBC-INT FD estimators are not limited to just one type of images. Conversely, they rather tend to extend to the considerable majority of the investigated non-synthetic input data.

8 AUXILIARY RESEARCH ON BOX-COUNTING

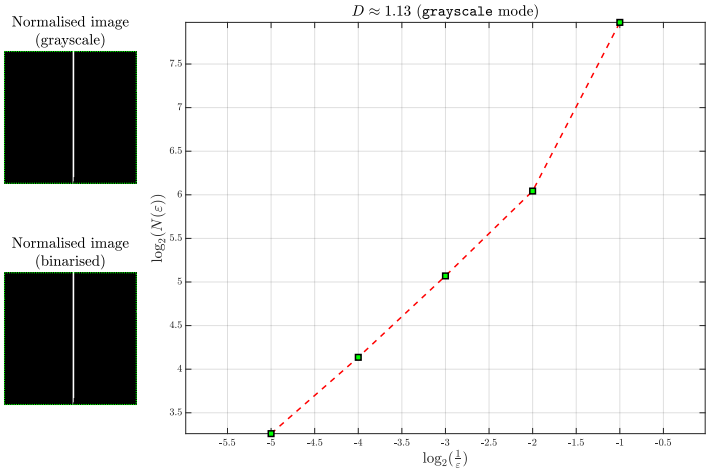
Figures 8.1 and 8.2 present visualisations of FDE procedure. Note that due to mitigation of boundary effects by applying cut-off on extremities within box sizes sequence in fig. 8.1, the smallest box included into fractal dimension estimation (FDE) is at least of size 2 px × 2 px or larger such that the considered numbers of boxes do not exceed $(2^{10-1})^2 = 512^2 = 262144$ (due to the possibly very long time required to complete `max` operations on all boxes performed by MATLAB's `blockproc` functions when maximum-based variant of `resize` function is used for image downsampling). Similarly, the largest considered box size is a square of the side length being one eight of the input image side length so the minimal considered number of boxes is $(2^{2+1})^2 = 64$.

Figure 8.2 shows how omitting the aforementioned cut-off stage would affect linear regression in FDE algorithm and its result for input images from fig. 8.1. Particularly, in fig. 8.2b there are clearly visible deviations from straight line near extremes. These deviations does not occur in fig. 8.2a and fig. 8.2c, nevertheless, when setting a general use cut-off thresholds, it must be bear in mind that sacryfying some of the non-deviated extreme data points in such cases might pay off by reducing bias in deviation-affected ones. Thus, the thresholds in developed implementation have been chosen as a trade-off between retaining a reasonable amount of data used for linear regression and having the regression possibly unaffected by undesired boundary effects.

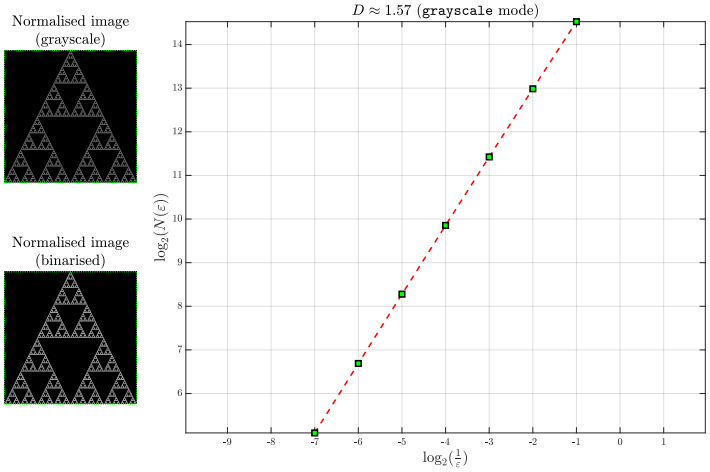
Figure 8.3 regards the effect of initial scaling (e.g., that might be applied by resizing during the pre-processing stage) of the input image on FDE process implemented utilizing weighted grayscale box-counting (W-GBC). The plots show how interpolation of the input data might affect the output fractal dimension (FD) estimate suggesting that there is a lower limit for the initial image size. This limit – in the mentioned cases – is about (500 to 1000) px. For smaller images, the value become noticeably biased. This in turn introduces a degree of unreliability into the result. In fig. 8.3a the border of this bias is sharp (just above the scale of 2) while in fig. 8.3c FD estimates lower rather gradually as the input image gets smaller. The bias observed when analyzing images that in this context might be considered *too small* is negative until they become much smaller. Fi-



(a) A thin straight section (`img1` shown in fig. 3.4a)

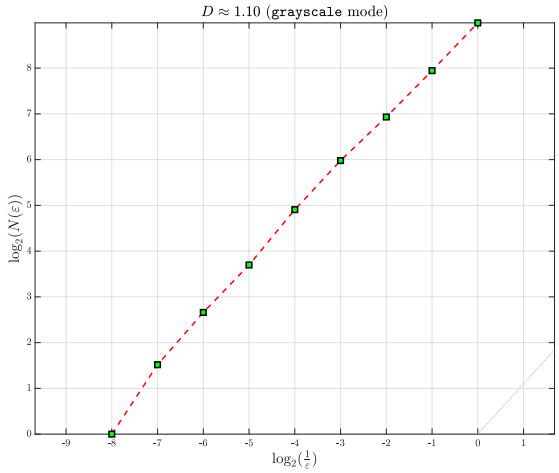


(b) A thick straight section with thinning lower ending surrounded by two single points on left and right image borders (`img3` shown in fig. 3.4c)

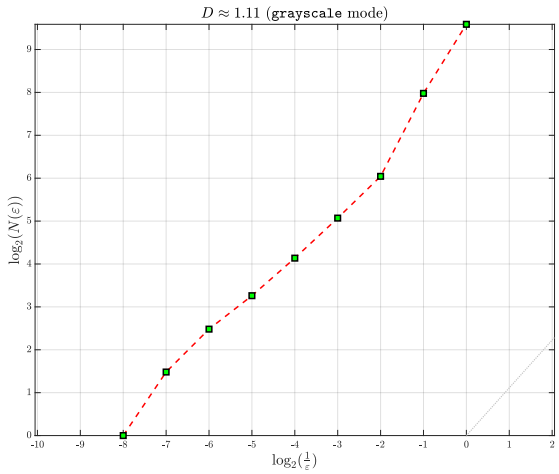


(c) Sierpinski's triangle in grayscale (`img11` shown in fig. 3.4k:)

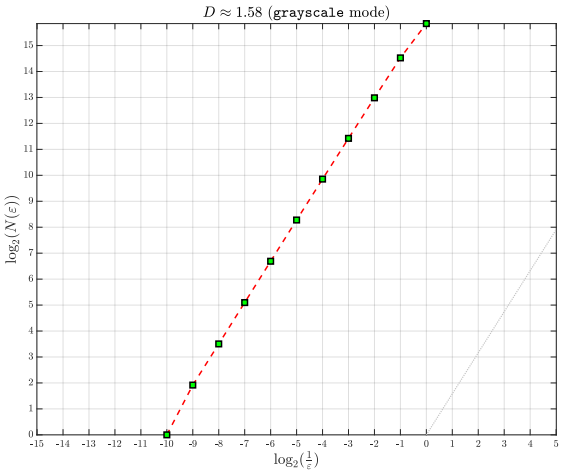
Figure 8.1: Illustrative FDE calculations visualized for individual synthetic images using W--GBC (note that the box sizes of extreme values have been cut off)



(a) A thin straight section (`img1` shown in fig. 3.4a)



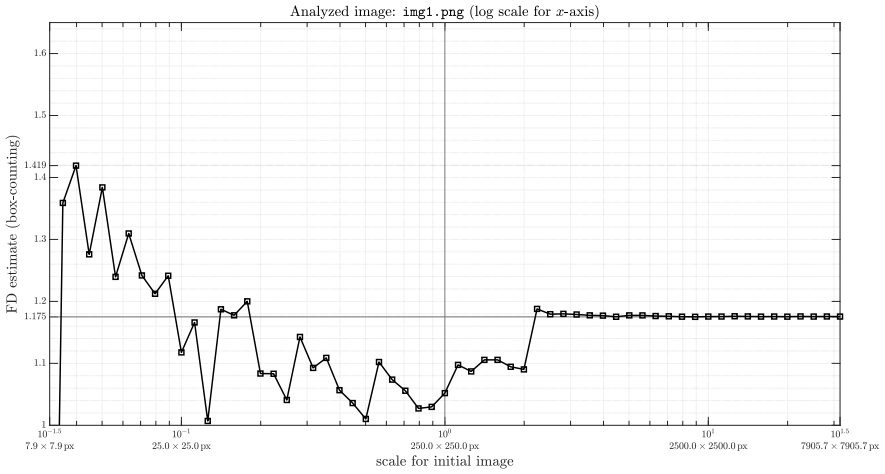
(b) A thick straight section with thinning lower ending surrounded by two single points on left and right image borders (`img3` shown in fig. 3.4c)



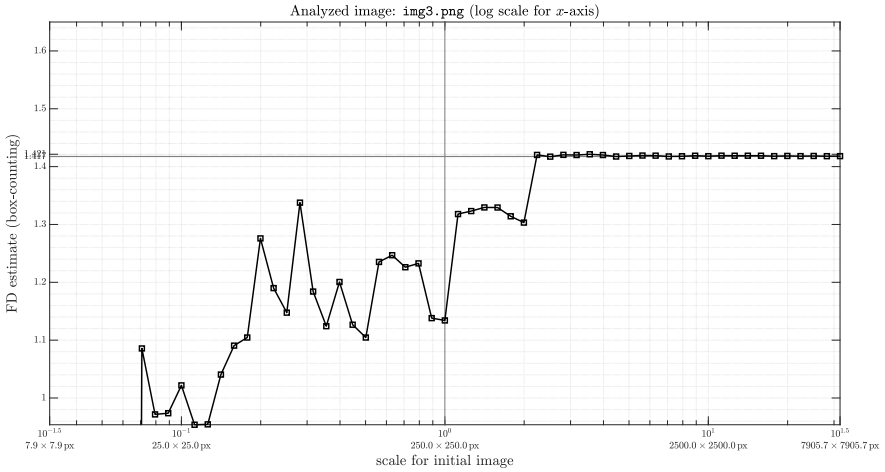
(c) Sierpinski's triangle in grayscale (`img11` shown in fig. 3.4k)

Figure 8.2: Illustrative FDE calculations visualized for individual synthetic images using W-GBC (note that in contrast to fig. 8.1 the cut-off of box sizes sequence has not been applied for demonstrative purposes)

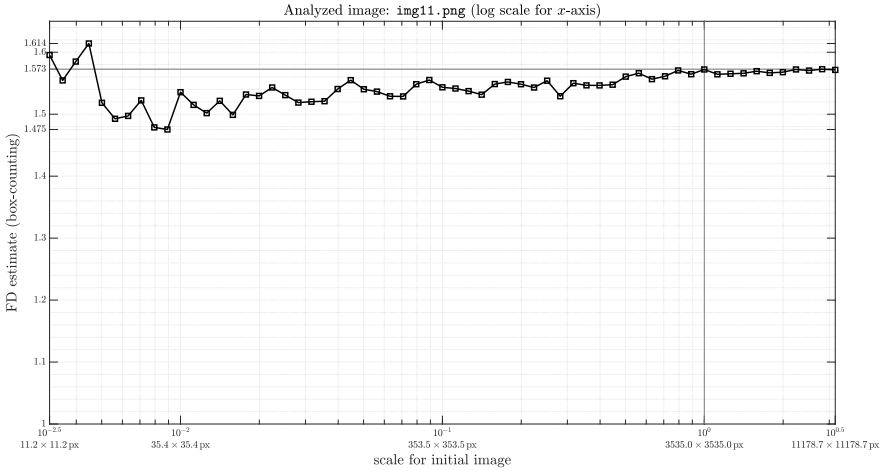
nally, for fig. 8.3a and fig. 8.3c as the image reach the side length of about (20 to 25) px the bias becomes positive. The observed absolute bias of FD estimates has been varying and reached values exceeding 0.4. Clearly, such level of uncertainty would be a non-negligible issue in many of the practical applications of FDE. Therefore, in a scenario when the large input image to be analyzed is to be divided into sub-frames, a care must be taken to ensure that their sizing is sufficient, e.g., by employing an initial upsampling if necessary.



(a) A thin straight section (`img1` shown in fig. 3.4a)



(b) A thick straight section with thinning lower ending surrounded by two single points on left and right image borders (`img3` shown in fig. 3.4c)



(c) Sierpinski's triangle in grayscale (`img11` shown in fig. 3.4k)

Figure 8.3: Illustrative analysis of the input image scale (size) influence on FDE using W--GBC for images from fig. 8.1

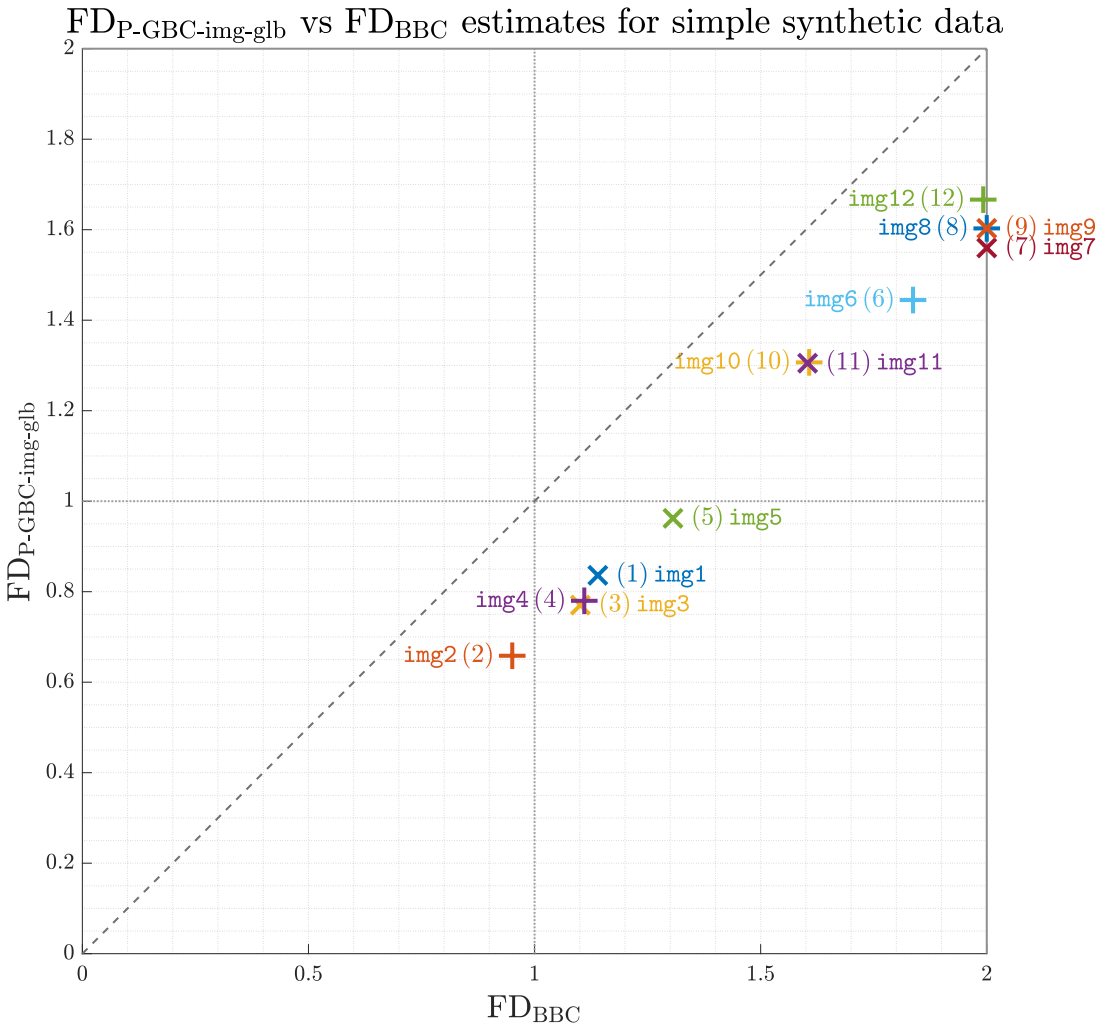
8.1 INTERESTING CASES

8.1.1 The interpretation of the results of fractal dimension estimation

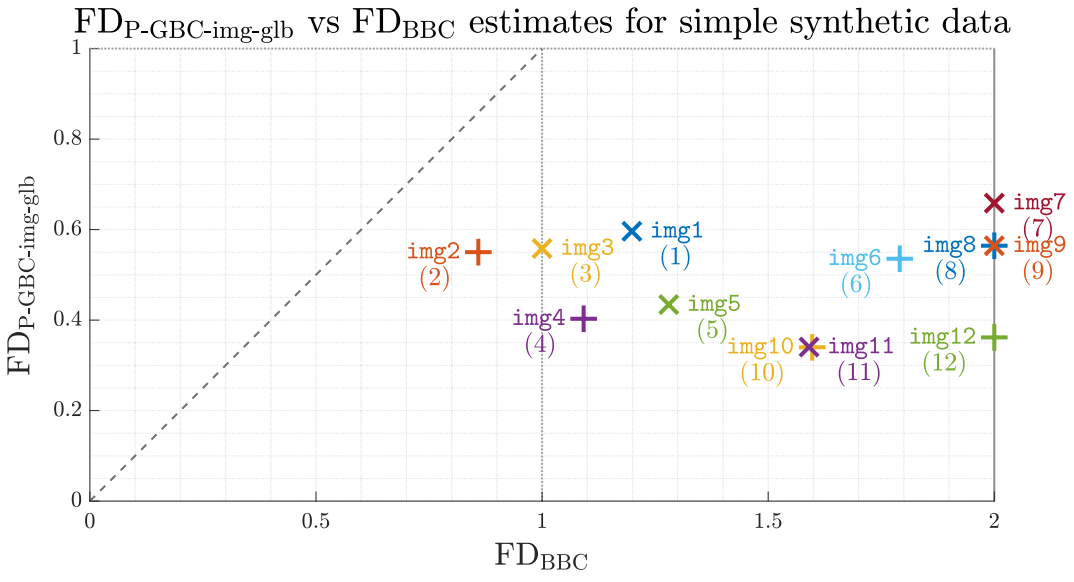
Figure 8.4 presents FD a comparison of estimates for $FD_{P\text{-}GBC\text{-}IMG\text{-}GLB}$ and FD_{BBC} estimators calculated for the set of 12 synthetic images useful for interpreting, assessing and validating the results of the estimation of more complex images of unknown or not well-defined (in terms of self-similarity) FD. The images used as synthetic input data are illustrated in fig. 3.4. To facilitate the visibility of the overlaying (`img8` and `img9`) or nearly overlaying (`img10` and `img11`) data points, odd-numbered images are marked with (diagonal) crosses and also labeled to the right of their markers while the even-numbered ones are marked with plus signs and labeled to the left of their markers.

The relation between P-GBC-IMG-GLB and BBC FD estimates is presented in two versions, namely fig. 8.4a and fig. 8.4b. A circumstance that has been responsible for the most prominent differences between these two implementations is the optimization of two important aspects of computation that has been reiterated multiple times. The first factor regards the fine-tuning of the parameters of box-counting (BC) downsampling function, q.v. section 3.2.4. The second one pertains to the optimization of the cut-off thresholds applied during the linear regression stage, q.v. section 3.1.2 and chapter 5. By comparing these two plots it may be noticed that the two mentioned factors play a vital – and sometimes even critical – role in achieving the results that are plausible in terms of reliability assured by overall coherence with the outputs of a standard method practiced in the field of interest, which in this case is binary BOX-COUNTING (BBC) algorithm.

To facilitate the presentation of the results, in fig. 8.4b, presenting an early implementation of FDE algorithm, axis of ordinates has been cropped to range $[0, 1]$, as no data point within this plot has an ordinate falling in range $[1, 2]$.



(a) The final implementation



(b) An early implementation, ordinates axis cropped to range [0, 1]

Figure 8.4: P-GBC-IMG-GLB and BBC FD estimates compared for synthetic data

Part V

SUMMARY

The following part discusses the results obtained it the work, summarizes them, and presents the drawn conclusions.

9 COMPARISON OF THE METHODS

9.1 FEATURES OF THE METHODS

An illustrative comparison of the features – including useful properties and limitations – provided by different numerical fractal dimension estimation (FDE) methods utilizing box-counting (BC) is presented in table 9.1. Note that in some cases the distinction between the full and partial support of a feature by a method is – – at least to some degree – heuristic. Nevertheless, diverging the level of the support into two classes may give the reader a better insight into the relative performance of two methods in cases where more than one of them supported a given property.

Mathematical accuracy in relation to the concept of *generalized dimensions* is described in chapter 4.

Consistency with analytical dimensionality measures touches the subject of self-similarity fractal dimension and Euclidean dimension likewise. They may be calculated theoretically for purely fractal (self-similar) sets and geometrical figures, respectively. It is the matter of discussion in chapter 1 and the results may be seen in fig. 3.5 within section 3.2.4.

Robustness relates to the subject of the simulation of artificial fringe patterns with a significant amount of noise. It is discussed in chapter 6, cf. figs. 6.1 and 6.2 and the results of this supplementary research is shown in figs. 7.6 and 7.6A together with fig. 7.7 in section 7.1.1 and figs. 8.1 to 8.3 in chapter 8.

Efficiency is elaborated on in section 7.1.2, cf. tables 7.4 and 7.5, in section 7.1.2, cf. fig. 7.12, and also – for older and not fully optimized implementations – in figs. 7.6 and 7.6A in section 7.1.1.

Paralellism is a subject raised in section 3.2.4 and in section 7.1.2, q.v. fig. 7.12. It applies mostly to weighted grayscale box-counting (W-GBC) and – intermediately – to binary box-counting (BBC) which may be implemented as a special case of W-GBC. Also applies to author's implementation for generating synthetic Saupe's fractals, nevertheless this is a input generation method that has not been investigated in reverse as a potential fractal dimension (FD) estimator.

Applicability to synthetic images regards the simple synthetic images presented in fig. 3.4 within section 3.2.4. The applicability to these images may be concluded on by referring, i.a., to fig. 3.5 in section 3.2.4 and fig. 8.4 in section 8.1.1.

Applicability to to artificial fractals constitutes a property focused on synthetic fractal data that is generated using Saupe's rescale-and-add (RAA)--based random fractal functions (RFFs). Examples are presented in fig. 5.1 in chapter 5 and the matter of efficiency is studied in the same section in figs. 5.2 and 5.3.

Applicability to USC-SIPI textures pertains to a class of images that is illustratively presented by fig. 7.15 within section 7.1.3 and is further commented on in the same section, cf. figs. 7.16 and 7.17.

Applicability to LSI video sequences regards the class of the images acquired in a in-vivo manner whose illustrative cases are shown in fig. 3.6 and also in figs. 7.2 and 7.3. An extended description of the methodological aspects is available in section 7.1.1, while performance of the methods in relation to interferometric data is described in section 3.2.4, q.v. figs. 3.7 and 3.8 and in section 7.1.2, q.v. tables 7.1 to 7.3 and fig. 7.11.

Table 9.1: Comparison of interferogram image descriptors exploiting box-counting-based fractal dimension estimation discussed in this work

		Box-counting methods			
		Binary	Grayscale		
			Probabilistic		*Weighted*
		BBC	P-GBC-img	P-GBC-int	W-GBC
Features	Mathematical accuracy	✓	★*	★*	×/—
	Consistency with analytical dimensions	✓	✓	×	✓
	Robustness on artificial and real data	×†	★/—	×/—	★
	Efficiency	✓‡	★	×	★
	Paralellism	✓‡	×	×	✓
	Applicability to synthetic images	✓§	★	×	✓
	Applicability to artificial fractals	✓§	✓	✓¶	★‖
	Applicability to USC-SIPI textures	★§	★	★	✓
	Applicability to LSI video sequences	★§	×	×	★

[✓] — full support.
[★] — partial support.
[×] — no support.
[—] — no data (lack of information).

* Fully mathematically accurate are P-GBC-img-glb and P-GBC-int-glb.
† Better results have been obtained for EBBC.
‡ As a special case of W-GBC.
§ Application of binary methods to grayscale (or multichannel) data is limited.
¶ Hindered by substantial computational overhead.
‖ Gives implausible results only for lacunarity coefficient $r = 4$.

DISCUSSION AND CONCLUSIONS

Several fractal-based image descriptors have been developed and investigated together with the other ones known in the literature and implemented by the author, sometimes with adding significant improvements to computational efficiency. As a mean of conducting reliable research, apart from obtaining real interferometric data, a method to automatically generate artificial interferograms has been developed as well as one of the best-rated methods in literature to generate artificial fractals [81] has been implemented and computationally improved.

One of the main goals of the work has been to construct fractal dimension (FD) estimators for two-dimensional grayscale images. They have been extensively studied and optimized in terms of numerical efficiency, high dynamic range of inputs as well as their consistency with self-similarity and Euclidean (topological) dimensions where they have been known. The second importand aim has regarded the applicability to classification tasks as a foundation to proposing novel diagnostic methods with application to a biomedical exploitations focused on dry eye syndrome (DES) detection or the assessment of the risk of this disease or possibly contact lens biocompatibility evaluation.

Some objectives have been postulated and taken into action to reach the postulated goals. Properties and limitation of the FD estimators in question have been studied theoretically and experimantally in the context of the available literature research coverage. A multi-criterial classification of the developed and existing methods has been established in terms of the form of the input data, e.g., grayscale vs binary, interpretation of the mathematical apparatus involved, e.g., probabilistic vs non-probabilistic, as well as technical properties regarding the image processing procedures applied, e.g., image-probability-space-based vs intensity-probability--space-based, local vs global etc.

Of the approaches studied for box-counting-based FRACTAL DIMENSION estimation there are two that have been observed to overcome some limitations of standard box-counting method. Namely, these fractal dimension estimators are weighted grayscale box-counting and probabilistic grayscale box-counting based on image probability space, of which the former is a novel contribution. Also an innovative addition is provided by the less efficient thus mathematically

accurate probabilistic grayscale box-counting based on intensity probability space (P-GBC-INT). These methods have been compared with the common binary box-counting approach with functions as a de facto benchmark for FRACTAL DIMENSION estimation (FDE). Other existing methods like differential box-counting (DBC) have been also included in the comparison.

Input data utilized during the study covers a wide range of images. The completeness of the empirical part of the doctoral work has been instituted by the means of starting with the simple synthetic figures for validation, continuing the research through the standard USC-SIPI textures and artificially disturbed synthesized interferograms and performing the final computations on the synthetic fractals generated using complex numerical tools as well as on the real data acquired with lateral shearing interferometry (LSI) technique on human subjects in-vivo.

One important result of the study is the development of new fractal dimension estimators that in some cases have been found to perform better than the known solutions. Another significant outcome is the improvements made in the field of methodology available for diagnosing DES. A theoretical apparatus has been developed to comprehensively describe the proposed mathematical and numerical solutions. Moreover, numerous comparisons to alternative methods (not only the fractal-based ones) in terms of efficiency, robustness, accuracy, applicability and numerical demand have been performed and their results provided after careful statistical analysis. New knowledge has been presented in the form of tables, plots, coherent ordering of terminology and classification of the methods, novel methodological approaches (including programmatic implementations of algorithms), practical comments and research conclusions. It has been show that the utility of the fractal-based methods for the investigated application could be no worse – and sometimes better – when compared with the existing solutions. The applicability potential that has been uncovered of legitimizes the statement that this field of study is of importance to some areas of industrial implementations and is a promising subject of further research, performed both in theoretical and practical directions.

Future research may encompass the computational optimization and fine-tuning of the numerous parameters of the developed methods to tailor them to particular practical applications in order to make their performance comparable to the currently applied solutions. Some theoretical work is also to be done, e.g., to fully cover the issue of resolving the mathematical formulations of some descriptors in question in order to ascertain whether they constitute measures of dimensionality in terms of the measure theory. Also, theoretical reducability and consistency linking the multitudinous empirical FDE-related descriptors and the self-similarity

dimension or other well-established measures of FD poses a task worth an extended analysis. Performing it would be a substantial scientific work that requires profound theoretical background and both mathematical and informatic experience, nevertheless it could be beneficial in terms of removing some flaws of the currently applied methods, finding better solutions to overcome their limitations (including completely new methods) as well as organizing, ordering and codifying the constantly expanding body of knowledge in this subject. Thus, the investigated area of study remains an open field for future studies.

One crucial inference that has been brought by this research is how important is the precise tailoring of the parameters of the method conjoined with the optimization of the pre-processing stage when a practical application of a fractal-based classifier is of interest.

The title of this work is "Diagnostic methods based on the analysis of interference images". Its general meaning is rather broad and covers a vast field of study covering numerous diagnostic solutions that are applied to extract the useful information from the interferometric data in various branches of science and technology, e.g., medicine, geography, industrial diagnostics, material fatigue detection, astronomy etc. The reason behind selecting such general term has been that at the time of the postulation of the subject the research performed for this thesis was at the preliminary stage. Conforming to the spirit of Huang [10] application of FD-based measures to classification tasks remains a delicate task. Namely, the matter of bringing together the mathematical accuracy of such descriptors with their practical utility might face a mutually exclusive criteria and thus is a matter of compromise that may be hard to settle for a particular combination of input data and chosen method. There is a multi-faceted spectrum of available fractal-based descriptors – – many of them still remaining novel and thus not investigated extensively like the intensity-and entropy-based grayscale variants. Moreover, there are many sub-families and branches of the basic methods caused by many parameters that are to be set to obtain a fractal descriptors, especially operating in grayscale or on a multi-channel images. Together with the numerousness of the available interferometric imaging techniques, this makes it virtually impossible to determine which FD-related descriptors and image acquisition techniques combination would be promising enough to be chosen for the final investigative study. The fractal-based methods have been chosen to overcome the limitations of the previously researched spectral techniques. It was initially intended to focus on the multifractality until the diversity and complexity of the still unexplored aspects of the regular fractal methods have driven the research to different track than predicted at the beginning. The gen-

erality of the chosen topic thus reflects the degree of uncertainty that inherently accompanies the research on fractal dimensionality. One of the biggest challenges and aims of such research is to possibly reduce its degree by revealing new information during the performed study and adding it to the body of this topic's knowledge afterwards.

DISSEMINATION OF THE RESULTS

The results obtained during the research conducted to complete the goals of the work described in this thesis have been disseminated as the successive progress has been occurring. Some of the initial results of the preliminary research have been assessed and compared in context of enhancing and extending the achievements of the previous research conducted by the author. It has been focused on applications of phase-based algorithms and morphological image analysis to extraction of the useful information from similar classes of images to those of interest in this work [108, 109]. Subsequently, as the body of the work has been expanding along with the theoretical and numerical apparatus, further publications have regarded their matter more in terms of comparing it with existing solutions of image diagnostics with the use of fractal dimension estimation (FDE).

Published works and the articles submitted for publication that regard the research described in this thesis are listed in the two sections below. They encompass journal and conference papers, respectively.

JOURNAL PAPERS

Serial articles published in scientific journals and periodicals include the following positions:

(1) P. D. Szyperski and D. R. Iskander, "New approaches to fractal dimension estimation with application to gray-scale images", *IEEE Access*, Sep. 8, 2019, submitted [62];
(2) P. D. Szyperski, "Comparative study on fractal analysis of interferometry images with application to tear film surface quality assessment", *Applied Optics*, vol. 57, no. 16, pp. 4491–4498, Jun. 1, 2018 [74].

CONFERENCE PAPERS

Papers published in proceedings of scientific conferences as well as talks and poster presentations associated with post-conference publications include the following positions:

(1) P. D. Szyperski and D. R. Iskander, "A novel phase-based approach to tear film surface quality assessment using lateral shearing interferometry", in *Proceedings of Computer Information Systems and Industrial Management, 14th IFIP TC 8 International Conference, CISIM* 2015, (Warsaw, Poland, Sep. 24–26, 2015), K. Saeed and W. Homenda, Eds., ser. Lecture Notes in Computer Science, vol. 9939, Cham, Switzerland, Heidelberg, Germany, New York, NY, USA, Dordrecht, the Netherlands, London, UK: Springer International Publishing, Oct. 30, 2015, pp. 435–447 [92];

(2) P. D. Szyperski and D. H. Szczęsna-Iskander, "Exploiting the phase domain of lateral shearing interferometry in measurements of tear film surface quality", in *Proceedings of VPOptics* 2014, *7th European Meeting coinciding with the 1st World Meeting in Visual and Physiological Optics* 2014, (Wrocław, Poland, Aug. 25–27, 2014), D. R. Iskander and H. Kasprzak, Eds., illustrator M. Jaskulski, World Meeting in Visual and Physiological Optics, Wrocław, Poland: Oficyna Wydawnicza Politechniki Wrocławskiej / Printing House of Wrocław University of Technology, Jun. 20, 2014, pp. 345–348 [108];

(3) P. D. Szyperski, *New algorithms for image analysis in non-invasive examination methods of corneal and contact lens tear film surface*, poster presented at Open Readings 2014: *57th* Scientific Conference for Young Students of Physics and Natural Sciences, Vilnius, Lithuania, Mar. 21, 2014 [110];

(4) P. D. Szyperski, *Nowe algorytmy analizy obrazów dla nieinwazyjnych metod badania filmu łzowego na oku i soczewkach kontaktowych [New algorithms for image analysis in non-invasive examination methods of corneal and contact lens tear film surface]*, Polish, poster presented at IV Ogólnopolska Konferencja Studencka Nowoczesne Metody Doświadczalne Fizyki, Chemii i Inżynierii 2013 [*4th* National Student Conference Modern Experimental Methods in Physics, Chemistry and Engineering 2013], Lublin, Poland / Lublin, Nov. 24, 2013 [111].

BIBLIOGRAPHY

[1] B. B. Mandelbrot, *The fractal geometry of nature*, B. B. Mandelbrot, Ed., ser. Einaudi paperbacks. Henry Holt and Company, 1983 (cit. on pp. 9, 16, 123).

[2] K. J. Falconer, *Fractal geometry, Mathematical foundations and applications*, 3rd ed. John Wiley & Sons, Feb. 2014 (cit. on pp. 9, 12).

[3] ——, *Techniques in fractal geometry*, 1st ed. John Wiley & Sons, Apr. 1997 (cit. on p. 9).

[4] ——, *The geometry of fractal sets*, H. Bass, H. Halberstam, J. F. C. Kingman, J. E. Roseblade and C. T. C. Wall, Eds., ser. Cambridge Tracts in Mathematics 85. Cambridge University Press, Jul. 1986 (cit. on p. 9).

[5] G. A. Edgar, *Measure, topology, and fractal geometry*, 2nd ed. S. J. Axler and K. A. Ribet, Eds., ser. Undergraduate Texts in Mathematics. New York, NY, USA: Springer Science+Business Media, LLC, Jan. 1, 2008, XVI, 272 pp. (cit. on p. 9).

[6] M. F. Barnsley and H. Rising, *Fractals everywhere*, M. F. Barnsley and H. Rising, Eds. Morgan Kaufmann, 1993 (cit. on pp. 9, 123).

[7] W. Kinsner, "A unified approach to fractal dimensions", in *Proceedings of the Fourth IEEE Conference on Cognitive Informatics*, IEEE, 2005, pp. 58–72 (cit. on pp. 9, 27, 123).

[8] J. L. Véhel, *INRIA/IRCCyN FracLab 2.1 product documentation*, 1998–2001 (cit. on p. 10).

[9] A. Karperien, *User's guide for FracLac, v. 2.5*, 2007–2012 (cit. on p. 10).

[10] Q. Huang, J. R. Lorch and R. C. Dubes, "Can the fractal dimension of images be measured?", *Pattern Recognition*, vol. 27, no. 3, pp. 339–349, 1994 (cit. on pp. 10, 27, 145).

[11] H.-O. Peitgen, H. Jürgens and D. Saupe, *Chaos and Fractals, New Frontiers of Science*, 2nd ed. New York, NY, USA: Springer-Verlag New York, 2004, XIII, 864 (cit. on pp. 11, 87).

[12] S. Wagon, *Mathematica® in Action, Problem Solving Through Visualization and Computation*, 3rd ed. New York, NY, USA: Springer-Verlag New York, 2010, p. 214, XI, 580 (cit. on p. 11).

[13] P. Łabędź , "Algorytmy fraktalne w cyfrowej analizie obiektów przestrzennych [Fractal algorithms in the digital analysis of spatial objects]", Polish, thesis, Politechnika Krakowska / Cracow University of Technology, Kraków / Cracow, Poland, May 24, 2016, 134 pp. (cit. on p. 11).

[14] M. Strzelecki and A. Materka, *Tekstura obrazów biomedycznych [Texture in biomedical images], Metody analizy komputerowej [Computer analysis methods]*, Polish, 1st ed. Warszawa / Warsaw, Poland: Wydawnictwo Naukowe PWN, Dec. 4, 2017, 224 pp. (cit. on p. 11).

[15] D. Schleicher, "Hausdorff dimension, its properties, and its surprises", *The American Mathematical Monthly*, vol. 114, no. 6, pp. 509–528, Jun. 2007 (cit. on p. 12).

[16] S. Peleg, J. Naor, R. Hartley and D. Avnir, "Multiple resolution texture analysis and classification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 4, pp. 518–523, 1984 (cit. on p. 12).

[17] N. Sarkar and B. B. Chaudhuri, "An efficient differential box-counting approach to compute fractal dimension of image", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 1, pp. 115–120, 1994 (cit. on pp. 12–17, 27, 37, 87, 106, 120).

[18] A. P. Pentland, "Fractal-based description of natural scenes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 661––674, 1984 (cit. on p. 12).

[19] J. M. Li, L. Lü, M. O. Lai and B. Ralph, *Image-based fractal description of microstructures*. Boston, MA, USA: Springer Science+Business Media, LLC, Apr. 17, 2013, XIII, 272 (cit. on p. 12).

[20] A. P. Pentland, "Shading into texture", *Artificial Intelligence*, vol. 29, no. 2, pp. 147–170, 1986 (cit. on pp. 13, 95).

[21] P. Brodatz, *Textures, A photographic album for designers and artists*, 1966 (cit. on p. 13).

[22] L.-Y. Wei, *The Brodatz textures are the standard testing set for texture analysis and synthesis. why don't you use more of them in your paper?*, S. C. G. Laboratory, Ed., 2003 (cit. on p. 13).

[23] J.-J. Gagnepain and C. Roques-Carmes, "Fractal approach to two--dimensional and three-dimensional surface roughness", *Wear*, vol. 109, no. 1, pp. 119–126, 1986 (cit. on pp. 13, 27).

[24] R. F. Voss, "Random fractals: Characterization and measurement", in *Scaling Phenomena in Disordered Systems*, ser. NATO Advanced Science Institutes B: Physics, R. Pynn and A. Skjeltorp, Eds., trans. by lass, 1st ed. Vol. 133, Springer Science+Business Media, LLC, Apr. 1991, ch. 1, pp. 1–11 (cit. on pp. 14, 19, 27).

[25] ——, "Characterization and measurement of random fractals", *Physica Scripta*, vol. 1986, no. T13, pp. 27–32, Jan. 1986 (cit. on p. 14).

[26] J. M. Keller, S. S. Chen and R. M. Crownover, "Texture description and segmentation through fractal geometry", *Computer vision, graphics, and image processing*, vol. 45, no. 2, pp. 150–166, 1989 (cit. on pp. 14, 27).

[27] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises and applications", *Society for Industrial and Applied Mathematics Review*, vol. 10, no. 4, pp. 422–437, 1968 (cit. on p. 14).

[28] J. Li, Q. Du and C. Sun, "A new box-counting method for estimation of image fractal dimension", in *Proceedings of the IEEE International Conference on Image Processing*, IEEE, 2006, pp. 3029–3032 (cit. on pp. 14, 17, 27, 106).

[29] S. S. Chen, J. M. Keller and R. M. Crownover, "On the calculation of fractal features from images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1087–1090, 1993 (cit. on pp. 14, 27).

[30] M. Long and F. Peng, "A box-counting method with adaptable box height for measuring the fractal feature of images", *Radioengineering*, vol. 22, no. 1, pp. 208–213, 2013 (cit. on p. 15).

[31] B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 72–77, 1995 (cit. on pp. 17, 27, 37, 106).

[32] S. Buczkowski, S. Kyriacos, F. Nekka and L. Cartilier, "The modified box-counting method: Analysis of some characteristic parameters", *Pattern Recognition*, vol. 31, no. 4, pp. 411–418, 1998 (cit. on pp. 17, 27).

[33] W.-S. Chen, S.-Y. Yuan and C.-M. Hsieh, "Two algorithms to estimate fractal dimension of gray-level images", *Optical Engineering*, vol. 42, pp. 2452--2464, 8 2003 (cit. on pp. 17, 27).

[34] J. Mishra and S. P. Pradhan, "Relative improved differential box-counting approach to compute fractal dimension of gray-scale image", College of Engineering and Technology, Tech. Rep., 2012 (cit. on p. 18).

[35] T. Dieker, "Simulation of fractional Brownian motion", Vrije Universiteit Amsterdam, Tech. Rep., 2004 (cit. on p. 18).

[36] M. Markou and S. Singh, "Novelty detection: A review", *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003 (cit. on p. 18).

[37] S.-C. Liu and S. Chang, "Dimension estimation of discrete-time fractional Brownian motion with applications to image texture classification", *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1176–1184, 1997 (cit. on p. 18).

[38] T. Ida and Y. Sambonsugi, "Image segmentation and contour detection using fractal coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 8, pp. 968–975, 1998 (cit. on pp. 18, 95).

[39] W. M. Krueger, S. D. Jost, K. Rossi and U. Axen, "On synthesizing discrete fractional Brownian motion with applications to image processing", *Graphical Models and Image Processing*, vol. 58, no. 4, pp. 334–344, 1996 (cit. on p. 18).

[40] D. P. Kroese and Z. I. Botev, "Spatial process generation", in *Lectures on Stochastic Geometry, Spatial Statistics and Random Fields*, V. Schmidt, Ed., ser. Lecture Notes in Mathematics. Springer Science+Business Media, LLC, Aug. 2014, vol. Volume II: Analysis, Modeling and Simulation of Complex Structures, ch. 12, pp. 369–404 (cit. on p. 18).

[41] M. L. Stein, "Fast and exact simulation of fractional Brownian surfaces", *Journal of Computational and Graphical Statistics*, vol. 11, no. 3, pp. 587–599, 2002 (cit. on p. 19).

[42] A. Napolitano, S. Ungania and V. Cannata, "Fractal dimension estimation methods for biomedical images", in *Matlab – A Fundamental Tool for Scientific Computing and Engineering Applications*, V. N. Katsikis, Ed., vol. 3, Rijeka, Croatia: InTech, 2012, ch. 7, p. 161 (cit. on p. 19).

[43] D. da Silva, F. Boudon, C. Godin, O. Puech, C. Smith and H. Sinoquet, "A critical appraisal of the box counting method to assess the fractal dimension of tree crowns", in *Advances in Visual Computing*, G. Bebis, R. D. Boyle, B. Parvin, D. Koračin, P. Remagnino, A. V. Nefian, G. Meenakshisundaram, V. Pascucci, J. Zara, J. Molineros,

H. Theisel and T. Malzbender, Eds., Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2006, pp. 751–760 (cit. on p. 19).

[44] M. Ivanovici and N. Richard, "Fractal dimension of color fractal images", *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 227–235, 2011 (cit. on pp. 19, 87).

[45] H.-O. Peitgen, D. Saupe and R. F. Voss, *The Science of Fractal images*, 1st ed. H.-O. Peitgen and D. Saupe, Eds. Springer-Verlag New York, 1988 (cit. on pp. 19, 31).

[46] W. Kasprzak, B. Lysik and M. Rybaczuk, *Dimensional analysis in the identification of mathematical models*, 1st ed. Singapore and Teaneck, NJ, USA, and Tottenridge, London, UK: World Scientific Publishing Co. Pte. Ltd., Sep. 1990, 204 pp. (cit. on p. 20).

[47] ——, *Measurements, dimensions, invariant models and fractals*, 1st ed. Wrocław, Poland and Lviv, Ukraine: Wrocław University of Technology: Ukrainian Society on Fracture Mechanics: Spolom, 2004, 300 pp. (cit. on p. 20).

[48] Google LLC. (Aug. 8, 2019). Google Scholar, "fractal dimension" and 'biomedical', bibliographic database search query, Google LLC, URL: https://scholar.google.com/scholar?q="fractal+dimension"+biomedical (visited on Aug. 8, 2019) (cit. on p. 20).

[49] C. Llorens-Quintana and D. R. Iskander, "Assessment of tear film using videokeratoscopy based on fractal dimension", *Optometry and Vision Science*, vol. 95, no. 2, pp. 32–42, Feb. 1, 2019 (cit. on p. 20).

[50] B. R. Masters, "Fractal analysis of the vascular tree in the human retina", *Annual Review of Biomedical Engineering*, vol. 6, no. 1, pp. 427–452, Apr. 13, 2004 (cit. on p. 20).

[51] G. Liew, J. J. Wang, N. Cheung, Y. P. Zhang, W. Hsu, M. L. Lee, P. Mitchell, G. Tikellis, B. Taylor and T. Y. Wong, "The retinal vasculature as a fractal, Methodology, reliability, and relationship to blood pressure", *Ophthalmology*, vol. 115, no. 11, pp. 1951–1956, Nov. 1, 2008 (cit. on p. 21).

[52] F. Family, B. R. Masters and D. E. Platt, "Fractal pattern formation in human retinal vessels", *Physica D: Nonlinear Phenomena*, vol. 38, no. 1–3, pp. 98–103, Sep. 1, 1989 (cit. on p. 21).

[53] M. A. Mainster, "The fractal properties of retinal vessels, Embryological and clinical implications", *Eye*, vol. 4, no. 1, pp. 235–241, Jan. 1, 1990 (cit. on p. 21).

[54] S. Zahid, R. Dolz-Marco, K. B. Freund, C. Balaratnasingam, K. Dansingani, F. Gilani, N. Mehta, E. Young, M. R. Klifto, B. Chae, L. A. Yannuzzi and J. A. Young, "Fractal dimensional analysis of optical coherence tomography angiography in eyes with diabetic retinopathy", *Investigative Ophthalmology & Visual Science*, vol. 57, no. 11, pp. 4940–4947, Sep. 1, 2016 (cit. on p. 21).

[55] S. G. K. Gadde, N. S. Anegondi, D. Bhanushali, L. Chidambara, N. K. Yadav, A. Khurana and A. S. Roy, "Quantification of vessel density in retinal optical coherence tomography angiography images using local fractal dimension", *Investigative Ophthalmology & Visual Science*, vol. 57, no. 1, pp. 246–252, Jan. 1, 2016 (cit. on p. 21).

[56] F. Corvi, M. Pellegrini, S. Erba, M. Cozzi, G. Staurenghi and A. Giani, "Reproducibility of vessel density, fractal dimension, and foveal avascular zone using 7 different optical coherence tomography angiography devices", *American Journal of Ophthalmology*, vol. 186, pp. 25–31, Feb. 1, 2018 (cit. on p. 21).

[57] S. Iannaccone, Y. Zhou, D. Walterhouse, G. Taborn, G. Landini and P. Iannaccone, "Three dimensional visualization and fractal analysis of mosaic patches in rat chimeras, Cell assortment in liver, adrenal cortex and cornea", *PLOS One*, vol. 7, no. 2, e31609, D. S. Milstone, Ed., pp. 1–14, Feb. 7, 2012 (cit. on p. 21).

[58] Ş. Ţălu, S. Stach, V. Sueiras and N. M. Ziebarth, "Fractal analysis of AFM images of the surface of Bowman's membrane of the human cornea", *Annals of Biomedical Engineering*, vol. 43, no. 4, pp. 906–916, Sep. 30, 2014 (cit. on p. 21).

[59] A. Karperien, H. Ahammer and H. F. Jelinek, "Quantitating the subtleties of microglial morphology with fractal analysis", *Frontiers in Cellular Neuroscience*, vol. 7, Article 3, pp. 1–18, Jan. 30, 2013 (cit. on p. 21).

[60] L. S. Liebovitch, J. Fischbarg and J. P. Koniarek, "Ion channel kinetics, A model based on fractal scaling rather than multistate markov processes", *Mathematical Biosciences*, vol. 84, no. 1, pp. 37–68, May 1, 1987 (cit. on p. 21).

[61] S. S. Cross, "Fractals in pathology", *The Journal of Pathology*, vol. 182, no. 1, pp. 1–8, Apr. 19, 1997 (cit. on p. 21).

[62] P. D. Szyperski and D. R. Iskander, "New approaches to fractal dimension estimation with application to gray-scale images", *IEEE Access*, Sep. 8, 2019, submitted (cit. on pp. 27, 124, 147).

[63] X. C. Jin, S. H. Ong and Jayasooriah, "A practical method for estimating fractal dimension", *Pattern Recognition Letters*, vol. 16, no. 5, pp. 457–464, 1995 (cit. on p. 27).

[64] E. Skubalska-Rafajłowicz, "A new method of estimation of the box-counting dimension of multivariate objects using space-filling curves", *Nonlinear Analysis: Theory Methods & Applications*, vol. 63, no. 5, e1281–e1287, 2005, Invited Talks from the Fourth World Congress of Nonlinear Analysts (WCNA 2004) (cit. on p. 27).

[65] J. Li, Q. Du and C. Sun, "An improved box-counting method for image fractal dimension estimation", *Pattern Recognition*, vol. 42, no. 11, pp. 2460–2469, 2009 (cit. on p. 27).

[66] M. Fernández-Martínez and M. A. Sánchez-Granero, "How to calculate the Hausdorff dimension using fractal structures", *Applied Mathematics and Computation*, vol. 264, pp. 116–131, 2015 (cit. on p. 27).

[67] J. Golay and M. Kanevski, "A new estimator of intrinsic dimension based on the multipoint morisita index", *Pattern Recognition*, vol. 48, no. 12, pp. 4070–4081, 2015 (cit. on p. 27).

[68] A. R. Backes and O. M. Bruno, "Texture analysis using volume-radius fractal dimension", *Applied Mathematics and Computation*, vol. 219, no. 11, pp. 5870–5875, 2013 (cit. on p. 27).

[69] A. R. Backes, "Upper and lower volumetric fractal descriptors for texture classification", *Pattern Recognition Letters*, vol. 92, pp. 9–16, 2017 (cit. on p. 27).

[70] K. Foroutan-pour, P. Dutilleul and D. L. Smith, "Advances in the implementation of the box-counting method of fractal dimension estimation", *Applied Mathematics and Computation*, vol. 105, no. 2–3, pp. 195–210, Nov. 1, 1999 (cit. on p. 27).

[71] A. R. Backes and O. M. Bruno, "A new approach to estimate fractal dimension of texture images", in *Image and Signal Processing*, A. Elmoataz, O. Lezoray, F. Nouboud and D. Mammass, Eds., Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2008, pp. 136–143 (cit. on p. 27).

[72] D. H. Szczęsna, J. Jarooński, H. T. Kasprzak and U. Stenevi, "Interferometric measurements of dynamic changes of tear film", *Journal of Biomedical Optics*, vol. 11, no. 3, pp. 1–8, 034028 2006 (cit. on pp. 27, 95).

[73] D. H. Szczęsna, D. Alonso-Caneiro, D. R. Iskander, S. A. Read and M. J. Collins, "Predicting dry eye using noninvasive techniques of tear film surface assessment", *Investigative Ophthalmology & Visual Science*, vol. 52, no. 2, pp. 751–756, 2011 (cit. on pp. 27, 95, 97).

[74] P. D. Szyperski, "Comparative study on fractal analysis of interferometry images with application to tear film surface quality assessment", *Applied Optics*, vol. 57, no. 16, pp. 4491–4498, Jun. 1, 2018 (cit. on pp. 27, 95, 147).

[75] A. R. Backes and O. M. Bruno, "Fractal and multi-scale fractal dimension analysis: A comparative study of Bouligand–minkowski method", *INFOCOMP Journal of Computer Science*, vol. 7, no. 2, pp. 74–83, Jun. 1, 2008 (cit. on p. 27).

[76] M. Cross, Class lecture, topic: "Introduction to Chaos", Part 2: Diagnostics, Chapter 9: Dimensions, California Institute of Technology, http://www.cmp.caltech.edu/~mcc/Chaos_Course/Lesson9/Dimension.pdf, Physics 161, Pasadena, CA, USA, Mar. 6, 2000 (cit. on pp. 37, 61).

[77] A. Block, W. von Bloh and H. J. Schellnhuber, "Efficient box-counting determination of generalized fractal dimensions", *Physical Review A. Atomic, molecular, and optical physics*, vol. 42, no. 4, pp. 1869–1874, Aug. 1990 (cit. on pp. 37, 61).

[78] H. G. E. Hentschel and I. Procaccia, "The infinite number of generalized dimensions of fractals and strange attractors", *Physica D: Nonlinear Phenomena*, vol. 8, no. 3, pp. 435–444, Sep. 1983 (cit. on p. 61).

[79] A. N. Kolmogorov, "Grundbegriffe der Wahrscheinlichkeitrechnung [Foundations of probability]", trans. from the German by N. Morrison, *Ergebnisse der Mathematik und ihrer Grenzgebiete*, vol. 2, no. 3, 1, Zentralblatt für Mathematiker, Ed., pp. 1–62, 1933–1950, Berlin–Heidelberg: Springer-Verlag New York: Chelsea Publishing Company, 1950 (cit. on p. 66).

[80] G. Shafer and V. Vovk, "The origins and legacy of kolmogorov's Grundbegriffe", *The Game – Theoretic Probability and Finance Project Working Paper 04*, arXiv:1802.06071, pp. 1–104, Feb. 5, 2018 (cit. on p. 66).

[81] D. Saupe, "Point evaluation of multi-variable random fractals", in *Visualisierung in Mathematik und Naturwissenschaften*, H. Jürgens and D. Saupe, Eds., ser. Bremer Computergraphik-Tage 1988. Berlin–Heidelberg: Springer-Verlag, Jan. 1989, pp. 114–126 (cit. on pp. 77–82, 84, 88, 143).

[82] R. E. Plotnick, R. H. Gardner, W. W. Hargrove, K. Prestegaard and M. Perlmutter, "Lacunarity analysis, A general technique for the analysis of spatial patterns", *Physical Review E*, vol. 53, pp. 5461–5468, 5 May 1, 1996 (cit. on pp. 77, 82).

[83] A. Fournier, D. S. Fussell and L. C. Carpenter, "Computer rendering of stochastic models", *Communications of the ACM*, vol. 25, no. 6, pp. 371–384, Jun. 1, 1982 (cit. on p. 87).

[84] F. K. Musgrave, "Procedural fractal terrains", in *Texturing and Modeling, A Procedural Approach*, ser. The Morgan Kaufmann Series in Computer Graphics, D. S. Ebert, Ed., 3rd ed. San Francisco, CA, USA: Elsevier Science (USA), Jan. 1, 2003, ch. 16, pp. 489–490 (cit. on p. 88).

[85] I. Marák. (Apr. 15, 1997). On synthetic terrain erosion modeling, A survey, URL: http://cescg.org/CESCG97/marak/index.html (visited on May 30, 2018) (cit. on p. 88).

[86] The Geometry Center, University of Minnesota. (Feb. 11, 1997). General Interest : Fractals : Clouds, Cloud simulations using spectral synthesis, description of generated images, University of Minnesota, URL: http://www.geom.uiuc.edu/graphics/pix/General_Interest/Fractals/Clouds/ (visited on Aug. 14, 2019) (cit. on p. 88).

[87] B. Remeseiro López, M. Penas, N. Barreira Rodríguez, A. Mosquera González, J. Novo and C. García Resúa, "Automatic classification of the interferential tear film lipid layer using colour texture analysis", *Computer Methods and Programs in Biomedicine*, vol. 111, no. 1, pp. 93–103, 2013 (cit. on p. 95).

[88] L. Ramos García, N. Barreira Rodríguez, A. Mosquera González, M. F. G. Penedo, E. Yebra-Pimentel Vilar and C. García Resúa, "Analysis of parameters for the automatic computation of the tear film break-up time test based on CCLRU standards", *Computer Methods and Programs in Biomedicine*, vol. 113, no. 3, pp. 715–724, 2014 (cit. on p. 95).

[89] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural features for image classification", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973 (cit. on p. 95).

[90] T. Chuen-Lin, L. You-Ru and J. Shiao-Shan, "Surface flatness of optical thin films evaluated by gray level co-occurrence matrix and entropy", *Applied Surface Science*, vol. 254, no. 15, pp. 4762–4767, May 30, 2008 (cit. on p. 95).

[91] D. H. Szczęsna and D. R. Iskander, "Lateral shearing interferometry for analysis of tear film surface kinetics", *Optometry and Vision Science*, vol. 87, no. 7, pp. 513–517, Jul. 2010 (cit. on pp. 95, 97, IV).

[92] P. D. Szyperski and D. R. Iskander, "A novel phase-based approach to tear film surface quality assessment using lateral shearing interferometry", in *Proceedings of Computer Information Systems and Industrial Management, 14th IFIP TC 8 International Conference, CISIM 2015*, (Warsaw, Poland, Sep. 24–26, 2015), K. Saeed and W. Homenda, Eds., ser. Lecture Notes in Computer Science, vol. 9939, Cham, Switzerland, Heidelberg, Germany, New York, NY, USA, Dordrecht, the Netherlands, London, UK: Springer International Publishing, Oct. 30, 2015, pp. 435–447 (cit. on pp. 95, 148).

[93] G. J. Ughi, K. Steigerwald, T. Adriaenssens, W. Desmet, G. Guagliumi, M. Joner and J. D'hooge, "Automatic characterization of neointimal tissue by intravascular optical coherence tomography", *Journal of Biomedical Optics*, vol. 19, no. 2, pp. 021 104–021 104, Jun. 24, 2013 (cit. on p. 95).

[94] D. H. Szczęsna, D. Alonso-Caneiro, D. R. Iskander, S. A. Read and M. J. Collins, "Lateral shearing interferometry, dynamic wavefront sensing, and high-speed videokeratoscopy for noninvasive assessment of tear film surface characteristics: A comparative study", *Journal of Biomedical Optics*, vol. 15, no. 3, pp. 1–9, 2010 (cit. on pp. 95, 105).

[95] D. Alonso-Caneiro, D. H. Szczęsna-Iskander, D. R. Iskander, S. A. Read and M. J. Collins, "Application of texture analysis in tear film surface assessment based on videokeratoscopy", *Journal Optometry*, vol. 6, no. 4, pp. 185–193, Aug. 28, 2013 (cit. on p. 95).

[96] U. R. Acharya, S. V. Sree, M. M. R. Krishnan, N. Krishnananda, S. Ranjan, P. Umesh and J. S. Suri, "Automated classification of patients with coronary artery disease using grayscale features from left ventricle

echocardiographic images", *Computer Methods and Programs in Biomedicine*, vol. 112, no. 3, pp. 624–632, 2013 (cit. on p. 95).

[97] V. K. Shrivastava, N. D. Londhe, R. S. Sonawane and J. S. Suri, "Computer-aided diagnosis of psoriasis skin images with HOS, texture and color features: A first comparative study of its kind", *Computer Methods and Programs in Biomedicine*, vol. 126, pp. 98–109, 2016 (cit. on p. 95).

[98] S. D. Tzikopoulos, M. E. Mavroforakis, H. V. Georgiou, N. Dimitropoulos and S. Theodoridis, "A fully automated scheme for mammographic segmentation and classification based on breast density and asymmetry", *Computer Methods and Programs in Biomedicine*, vol. 102, no. 1, pp. 47–63, 2011 (cit. on p. 95).

[99] D. H. Szczęsna, H. T. Kasprzak, J. Jaroński, A. Rydz and U. Stenevi, "An interferometric method for the dynamic evaluation of the tear film", *Acta Ophthalmologica Scandinavica*, vol. 85, no. 2, pp. 202–208, Nov. 17, 2006 (cit. on p. 96).

[100] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density", trans. from the Russian by B. Seckler, *Theory of Probability and Its Applications*, vol. 14, no. 1, pp. 153–158, Feb. 1, 1967, Society for Industrial and Applied Mathematics (cit. on pp. 99, XI).

[101] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1, 1979 (cit. on p. 100).

[102] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation", *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–165, Jan. 1, 2004 (cit. on p. 100).

[103] A. J. Lipiński and S. Lipiński, "Automatyczna ocena jakości oprysku na podstawie śladów kropel przy użyciu komputerowej analizy obrazu [Automatic valuation of the spraying quality on the basis of drops traces using the computer image analysis]", Polish, *Inżynieria Rolnicza / Agricultural Engineering*, vol. 5, no. 114, pp. 163–168, Oct. 21, 2009 (cit. on p. 100).

[104] D. Bradley and G. Roth, "Adaptive thresholding using the integral image", *Journal of Graphics Tools*, vol. 12, no. 2, pp. 13–21, Jul. 1, 2007 (cit. on p. 100).

[105] The MathWorks®, Inc., *MATLAB® 9.7 R2019b Prerelease product documentation*, Natick, MA, USA, 1994–2019 (cit. on p. 100).

[106] P. D. Welch, "The use of FFT for the estimation of power spectra: A method based on time averaging over short, modified periodograms", *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, F. T. Van Vee, Ed., pp. 70–73, Jun. 1967 (cit. on p. 122).

[107] USC Viterbi School of Engineering. (Jan. 1, 1977). The USC-SIPI image database, Volume 1: Textures. A. Weber, Ed. 155 monochrome images (130 images of size 512 × 512 px and 25 images of size 1024 × 1024 px), URL: http://sipi.usc.edu/database/database.php?volume=textures (visited on Feb. 28, 2018) (cit. on pp. 125–127).

[108] P. D. Szyperski and D. H. Szczęsna-Iskander, "Exploiting the phase domain of lateral shearing interferometry in measurements of tear film surface quality", in *Proceedings of VPOptics 2014, 7th European Meeting coinciding with the 1st World Meeting in Visual and Physiological Optics 2014*, (Wrocław, Poland, Aug. 25–27, 2014), D. R. Iskander and H. Kasprzak, Eds., illustrator M. Jaskulski, World Meeting in Visual and Physiological Optics, Wrocław, Poland: Oficyna Wydawnicza Politechniki Wrocławskiej / Printing House of Wrocław University of Technology, Jun. 20, 2014, pp. 345–348 (cit. on pp. 147, 148).

[109] P. D. Szyperski, "Enhancing the optical capabilities of the lateral shearing interferometry [Rozwinięcie optycznego działania interferometru typu lateral shearing do pomiaru filmu łzowego na oku]", Master's thesis, Wrocław University of Science and Technology / Politechnika Wrocławska, Wrocław, Poland / Wrocław, Jul. 23, 2013 (cit. on p. 147).

[110] ——, *New algorithms for image analysis in non-invasive examination methods of corneal and contact lens tear film surface*, poster presented at Open Readings 2014: 57th Scientific Conference for Young Students of Physics and Natural Sciences, Vilnius, Lithuania, Mar. 21, 2014 (cit. on p. 148).

[111] ——, *Nowe algorytmy analizy obrazów dla nieinwazyjnych metod badania filmu łzowego na oku i soczewkach kontaktowych [New algorithms for image analysis in non-invasive examination methods of corneal and contact lens tear film surface]*, Polish, poster presented at IV Ogólnopolska Konferencja Studencka Nowoczesne Metody Doświadczalne Fizyki, Chemii i Inżynierii 2013 [4th National Student Conference Modern Experimental Methods in Physics, Chemistry and Engineering 2013], Lublin, Poland / Lublin, Nov. 24, 2013 (cit. on p. 148).

[112] R. J. Oosterbaan, "Frequency and regression analysis", in *Drainage Principles and Applications*, 16, H. P. Ritzema, Ed., 2nd revised edition, part 2, Wageningen, the Netherlands: International Institute for Land Reclamation and Improvement (ILRI), Jun. 1, 1994, ch. 6, pp. 175–224 (cit. on p. III).

[113] S. S. Wilks, "The large-sample distribution of the likelihood ratio for testing composite hypotheses", *The Annals of Mathematical Statistics*, vol. 9, no. 1, pp. 60–62, Mar. 1, 1938 (cit. on p. III).

[114] S. Glover and P. Dixon, "Likelihood ratios, A simple and flexible statistic for empirical psychologists", *Psychonomic Bulletin & Review*, vol. 11, no. 5, pp. 791–806, Oct. 1, 2004 (cit. on p. III).

[115] GraphPad Software Inc. (Apr. 25, 2010). Segmental linear regression, Knowledgebase – article #1101, URL: https://www.graphpad.com/support/faq/segmental-linear-regression/ (visited on Dec. 22, 2018) (cit. on p. IV).

[116] J. Fogarty, R. S. Baker and S. E. Hudson, "Case studies in the use of ROC curve analysis for sensor-based estimates in human computer interaction", in *Proceedings of Graphics Interface 2005*, ser. GI '05, Victoria, BC, Canada: Canadian Human-Computer Communications Society, May 9, 2005, pp. 129–136 (cit. on p. IX).

[117] T. Hastie, R. Tibshirani and J. Friedman, *The elements of statistical learning, Data mining, inference, and prediction*, 2nd ed. Ser. Springer Series in Statistics. New York, NY: Springer New York, Feb. 1, 2009, XXII, 745 (cit. on p. IX).

[118] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to acsroc, informedness, markedness and correlation", *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, Dec. 15, 2011 (cit. on pp. IX, X).

[119] P. Perruchet and R. Peereman, "The exploitation of distributional information in syllable processing", *Journal of Neurolinguistics*, vol. 17, no. 2, pp. 97–119, Mar. 1, 2004 (cit. on p. IX).

[120] W. J. Youden, "Index for rating diagnostic tests", *Cancer*, vol. 3, no. 1, pp. 32–35, Jan. 1, 1950 (cit. on pp. IX, X).

[121] E. F. Schisterman, N. J. Perkins, A. Liu and H. Bondell, "Optimal cut-point and its corresponding Youden index to discriminate individuals using pooled blood samples", *Epidemiology*, vol. 16, no. 1, pp. 73–81, Jan. 1, 2005 (cit. on p. X).

[122] D. W. Scott, "On optimal and data-based histograms", *Biometrika*, vol. 66, no. 3, pp. 605–610, Dec. 1, 1979 (cit. on p. X).

[123] A. B. Tsybakov, *Introduction to Nonparametric Estimation*, 1st ed. New York, NY, USA: Springer-Verlag New York, Jan. 1, 2009, pp. 16–19, X, 214 (cit. on p. XI).

[124] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, 1st ed. D. R. Cox, D. V. Hinkley, N. Keiding, N. M. Reid, D. B. Rubin and B. W. Silverman, Eds., ser. Monographs on Statistics and Applied Probability 26. Boca Raton, FL, USA, London, UK, New York, NY, USA, Washington, D. C., USA: Chapman & Hall/CRC, Apr. 1, 1986, 176 pp., 1st CRC reprint 1998 (cit. on pp. XI, XII).

[125] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*, ser. Oxford Statistical Science Series. Oxford, United Kingdom: Oxford University Press, Jan. 1, 1997 (cit. on p. XII).

Appendix

A SEGMENTED LINEAR REGRESSION

Segmented regression, sometimes also referred to as piecewise regression, broken-stick regression or segmental regression, is an approach to regression analysis. It is based on partitioning of the independent variable into intervals and fitting a separate line segment to each of such formed segments [112].

A.1 BISEGMENTAL LINEAR REGRESSION

The system developed during this work for fractal dimension estimation (FDE) based on least squares linear regression slope utilizes an implementation of *bisegmental linear regression* – a special case of segmented linear regression. It is used for regression analysis of data that have a form similar to the one depicted in fig. A.1. Such segmented relations have been observed for some sequences of lateral shearing interferometry (LSI) images. In general, similar forms of characteristics of data also sometimes occur in analysis of dynamic changes in textural features as well as in other estimation-related tasks.

A *log-likelihood-ratio test* of whether a simple (unisegmental) linear fit is significantly worse than its bisegmental counterpart in terms of the goodness of fit as measured by coefficient of determination R^2. This test is based on *Samuel Stanley Wilks' theorem* that regards the matter of conditions under which the distribution of the doubled logarithm of the likelihood ratio converges to χ^2 distribution [113]. In tests performed during the research conducted for the purpose of this work, the null hypothesis has been that restricting to simple linear regression would not result in a fit whose goodness would have likelihood differing significantly from the likelihood of the goodness of fit provided by the bisegmental model. In detail, under the null hypothesis, both likelihoods should not differ by more than sampling error [114]. The significance level ((critical p-value)) used in the described work has been set to $\alpha = 0.01$. When the obtained p--value has been larger than or equal to α, the null hypothesis has been rejected, i.e., the model has been assumed to be characterized by the bisegmental form.

The common approach in the case of bisegmental linear regression is to utilize an unconstrained approach, but the one used in the work is conditional to prevent discontinuities. Illustrative realizations of unconstrained vs constrained variants of bisegmental linear regression have been contrasted with simple (unisegmental) linear regression and are presented in graphical form in fig. A.1. Both simple and bisegmental types of linear regression have operated on the same continuous bisegmental data with breakpoint at $x = 10$ (about one third of the range of the input variable). Data have been randomized with Gaussian noise prior to applying regression analysis. Standard deviation used for randomization of the right-hand-side segment has been larger than the used for left-hand-side one. The ratio of these standard deviations is proportional to the ratio of the lengths of their respective data range. That is, in the presented example, the right-hand-side data range is twice as long as its left-hand-side counterpart. Thus, it has been randomized with Gaussian noise of twice as large dispersion when compared to the standard deviation applied to the randomization of the other one.

Note that sample Pearson's correlation coefficient $r^2_{x,y}$ is slightly higher for the unconstrained bisegmental model shown in fig. A.1a than for the constrained model presented in fig. A.1b. This difference in favor of unconstrained models can sometimes be of much greater extent, as the fitting ability of statistical regression models grows with their generality. Another simple graphical example of segmental linear regression with some description regarding a practical application to scientific data analysis is available in [115].

For the research problem considered in this work, the trisegmental variant of segmented linear regression has been considered as a possible extension of the generality of the bisegmental type of regression. One reason behind this has been that constrained multisegmental linear regression has been applied with success as an analytical tool in related researches. The number of considered linear segments has been at least four, with an additional conclusion that for some cases of tear film surface kinetics assessment even up to five linear segments may pose a choice that is well-justified by the form of the experimental data [91]. This encourages the utilization of multisegmental linear regression with more than two segments. Nevertheless, for the majority of cases, it has rendered to be too demanding in terms of computational load when comparing them to potential benefits. Especially, the multisegmental approach to linear regression has not been strongly justified by the resultant FDE data form to be applied in the conducted numerical experiments. The computational demand of a variant of segmented linear regression with a given number of segments may be estimated by counting the total

number of complete regressions procedures required to be performed to chose the variant providing the best fit. Typically, the considered placings of the breakpoints in terms of the abscissæ are restricted to locations of the points in the data set. Then, this number is dependent on the number of the segments k and the number of the data points n being subject to the regression. Thus, it may be denoted as $N(n, k)$. The simplest way to obtain the value of $N(n, k)$ is to notice that, as the segment may be fitted reasonably fitted to at least two separate points, two extremal data points are to be excluded from the set of localizations of potential brakpoints. Thus, there are $n - 2$ candidate values for abscissæ of the breakpoints. Within them, k segments are to be placed, and to divide input range into k neighboring sub-ranges, $k-1$ breakpoints are required. The order in which the breakpoints are sorted is unimportant, as the segments are forced to be sorted in one distinguished order imposed by the natural non-decreasing sequence of the inputs of the estimated functional relation. Consequently, combination of $k - 1$ locations taken from $n - 2$ possible ones determine a single regression procedure. Therefore, the total number of regressions is equal to the number of these $k - 1$--combinations of $n - 2$-element set:

$$N(n, k) = \binom{n-2}{k-1}. \tag{A-1}$$

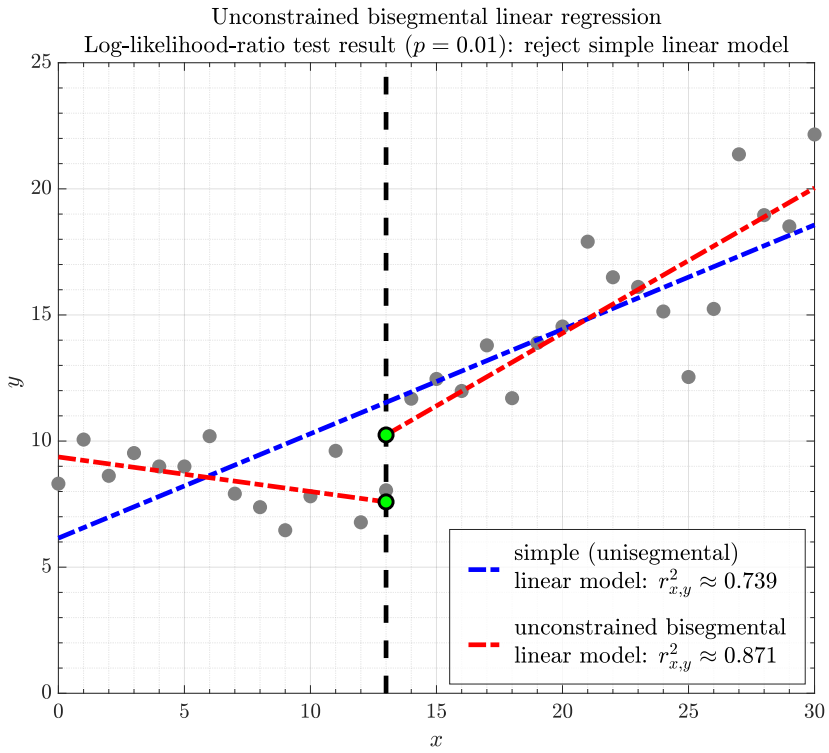
To obtain the optimal combination, each k-segmental linear regression of n data points requires $N(n, k)$ computations of the segmented linear regression algorithm, as stated by eq. (A-1). This means that the function of time complexity of such a regression optimization belongs to $\mathcal{O}(n^k)$ set:

$$N(n, k) \in \mathcal{O}(n^k). \tag{A-2}$$

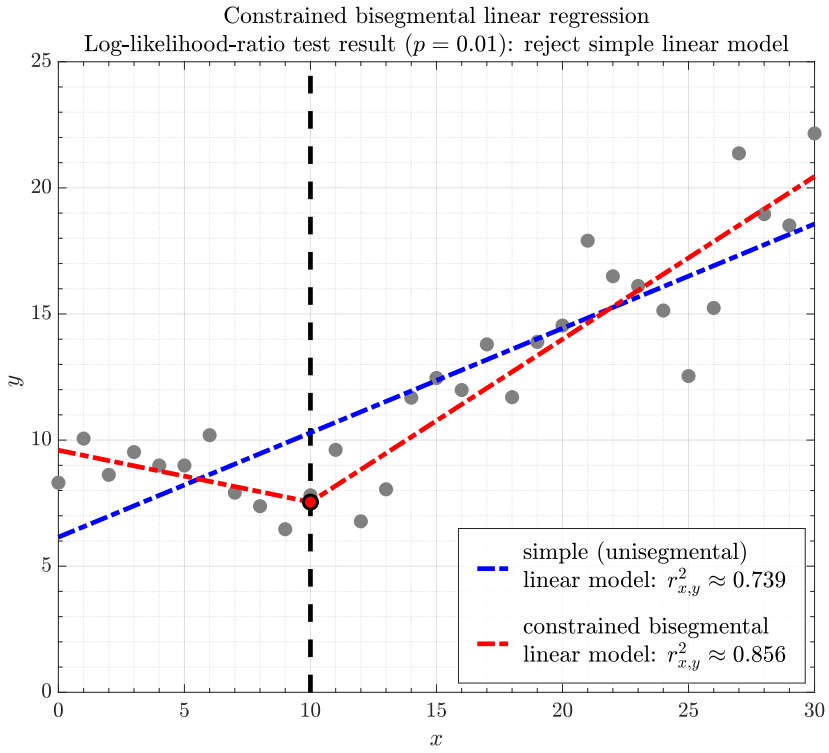
The substantial consequence of the type of computational time complexity formulated in eq. (A-2) is that the execution time of an algorithm fitting the breakup points for the multisegmental linear regression grows exponentially with the number of the segments to be fitted. The applicable aspect of this relation is that the choice of four or five segments instead of one or two in many cases results, e.g., in practical resignation from the near instantaneous or real-time computation for the minutes-lasting slowdown. For larger data sets that take many second to compute even in the simple linear regression variant the usage of trisegmental or more complex regression type may result in time overhead of multiple hours or days. Therefore, introducing multisegmental form when even the rough estimate of locations

of the breakpoints is unknown constitutes a decision that needs to be made with care if the profit of gaining potentially better fit outweighs the cost of the required additional processing time and computational power.

Listing C.5 demonstrates the core loop and fundamental calculations of the bisegmental linear regression implementation utilized within the research described in this work. Some additional parts of this function as well as the contents of its subroutines are excluded as they are based on conceptual and implementational solutions not developed by the author. Permission to utilize the mentioned portions of supplementary code for the purpose of the described numerical experiments has been granted, but it may not extend to the publication of the code itself. Additionally, the detailed descriptions of non-original technical contributions would be of minor interest in context of the purpose of this thesis.



(a) Unconstrained variant



(b) Constrained variant – used in the described research

Figure A.1: Illustrative plots of simple and segmental linear regression – a statistical tool utilized in this work for determination (numerical estimation) of FD of grayscale images.

B | RECEIVER OPERATING CHARACTERISTIC

Receiver operating characteristic (ROC) is a fundamental property of a detector or a classifier that describes the probability of detection $\mathbb{P}_D$ – also named sensitivity or true positives rate (TPR) – in terms of probability of false alarm $\mathbb{P}_{FA}$ – also termed – – 1 – specificity or false positives rate (FPR). ROC is often presented in a graphical form of a plot, i.e., as a curve.

Histogram-based approach is the most popular due to its mathematical simplicity as well as availability in majority of software statistical data analysis environments. It results in ROC curve in a form of a monotonic non-decreasing step function.

A number of parameters may be derived from ROC curve, including area under the curve (AUC) and Youden's J.

AUC, also termed A' [116] and the concordance statistic or – in short – c-statistic [117], is calculated by integrating ROC curve and is a number from $[0, 1]$ range describing the quality of the classifier evaluated by ROC. It may be interpreted as the probability that a classifier will rank a randomly chosen positive realization higher than a randomly chosen realization that is negative (in case when 'positive' ranks higher than 'negative').

Youden's J *statistic* is also termed *Youden's* J *index* and DeltaP′ and is in some contexts more or less closely related to statistical terms such as: informedness, inversed markedness, skew-insensitive quality function of novelty – also known as weighted relative accuracy (WRAcc) – together with INFOGAIN, Gini's coefficient (index) etc. [118, 119]. The index expresses an approach to describe a performance of a dichotomous diagnostic test in a simple manner by expressing it with single statistic. It is defined as [120]:

$$\mathrm{J} \doteq \text{sensitivity} + \text{specificity} - 1; \tag{B-1a}$$

what may be equivalently expanded to:

$$\mathrm{J} \doteq \frac{\mathrm{TPR}}{\mathrm{TPR} + \mathrm{FNR}} + \frac{\mathrm{TNR}}{\mathrm{TNR} + \mathrm{FPR}} - 1. \tag{B-1b}$$

Values of Youden's J index inclusively cover a range from zero to one [120]. The value reaches zero value when a diagnostic test gives the same proportion of positive results for both investigated groups. In such case the test is of no use. On the other hand, a value of one indicates that there are no false positives or false negatives. This is a case for an ideal test. Youden's J weights false positives and false negatives equally, therefore all tests resulting with the same value of this statistic give the same total proportion of misclassified results.

Youden's J statistic is frequently discussed in statistical context of assessing classifiers by means of ROC analysis [121]. While the index is defined for all points constituting ROC curve, its maximum value J_{max} is of special interest, as it may be used as a criterion for optimal cut-off determination for which the result of the test is not only dichotomous but numeric. J_{max} index is represented in plots in a visual manner as the height above the chance line (the increasing diagonal of the axes box). Its value is also equivalent of AUC delimited by a single operating point. While an individual value usually does not reveal as complete description of a test as the full ROC curve, this index is still useful in many applications. Particularly, it may be interpreted as an estimate of the probability of making an informed decision based on the classifier indication (with respect to a decision based on a random guess) taking into account all predictions. In a general case of multinomial (multiclass) tests, such estimate of an informed decision probability is defined under the name of *informedness*. In case of two classes these two measures, namely Youden's J and informedness, are equivalent [118].

As histogram-based approach results in step ROC curve that has discontinuities, the smoothness provided by kernel density estimation (KDE) may be preferred over the discreteness of the histogram. Also, kernel density estimates converge faster to the true underlying density for continuous random variables [122].

For a distribution with an unknown density f to be estimated and an univariate independent and identically distributed sample $(x_1, x_2, \ldots, x_n)$ drawn from that distribution, the *kernel density estimator* of f is:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{B-2}$$

where $K(\cdot)$ is kernel being a non-negative function and $h > 0$ is *bandwidth* also termed smoothing parameter. Moreover, $K_h(x) \doteq \frac{1}{h}K(\frac{x}{h})$ is referred to as scaled kernel. In general, to prevent over-smoothing h is minimized as much as the data

allows, but taking into consideration the *bias–variance trade-off* related to the issue of *bias-variance decomposition*.

Epanechnikov's kernel (alternatively termed *a parabolic kernel*) is defined as:

$$K(u) = \frac{3}{4}(1 - u^2), \tag{B-3}$$

where $|u| \leqslant 1$.

In terms of lowest possible asymptotic mean integrated squared error (AMISE), Epanechnikov's kernel is often described as optimal [100] and thus – despite some criticism [123] – remains a de facto standard [124] reference for comparing *kernel efficiency* in KDE. The eficciency of a kernel $K(u)$ is defined as:

$$\sqrt{\int u^2 K(u)\,du} \int (K(u))^2\,du. \tag{B-4}$$

Although Epanechnikov's kernel is referred to as optimal, difference in efficiency for some other kernels is small enough to neglect it in many practical applications. Moreover, due to favorable mathematical attributes, the *Gaussian kernel* (also known as *normal kernel*) exploiting the normal density function is sometimes preferred. It has the relative efficiency (with respect to Epanechnikov kernel) of about 95.1% and is defined as:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}, \tag{B-5}$$

where $u \in \mathbb{R}$.

Plots depicting both Epanechnikov's and Gaussian kernels are presented in fig. B.1.

Kernel estimation may cause the values to exceed the range of possible measures.

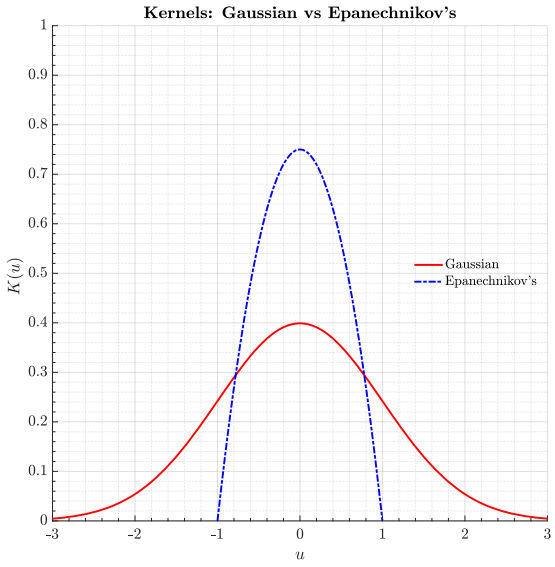


Figure B.1: Plot showing imposed graphs of Gaussian and Epanechnikov's kernels used for KDE of PDFs

B.1 IMPLEMENTATION OF CALCULATION AND ANALYSIS OF ROCS

B.1.1 ROCs calculation

Authors' implementation of ROC curves generator utilizes Epanechnikov's kernel.

Estimation is done in the range of gathered data. To overcome the clipping of estimates laying outside this interval, the resulting probability density function (PDF) is normalized to assure its integral being equal to one.

Bandwidth is determined as the optimal for estimating normal densities using the *normal distribution approximation* also referred to as *Gaussian approximation* or *Silverman's (1986) [124] rule-of-thumb* for bandwidth estimator. Normal basis functions are used to approximate univariate data, and the underlying density being estimated is assumed to be Gaussian. Thus, according to the aforementioned rule, the choice for bandwidth h minimizing mean integrated squared error (MISE), i.e., the optimal one [124, 125], is:

$$h = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}} \approx 1.06\hat{\sigma}n^{-1/5}, \tag{B-6}$$

where $\hat{\sigma}$ is the standard deviation of a sample.

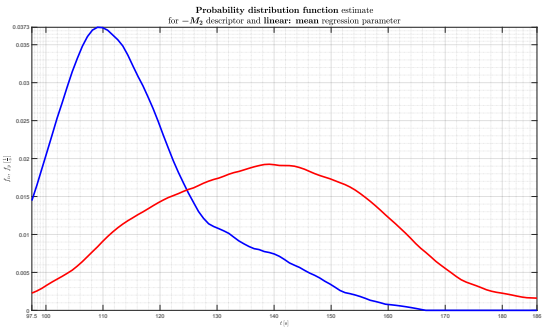
Probabilities estimated for ROCs are plotted as points while the continuous curve estimation for visualization purposes is plotted as an estimation based on best fit of a polynomial of fourth degree.

B.1.2 ROCs analysis

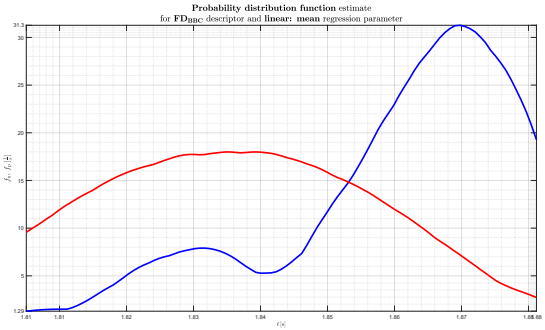
As a mean of ROC curves analysis, their AUCs have been calculated and displayed. Calculations have been performed with the exploitation of trapezoidal numerical integration. Moreover, also implemented has been calculation and visualisation of J_{max} values, i.e., maxima of Youden's J indices for ROCs.

B.1.3 Examples

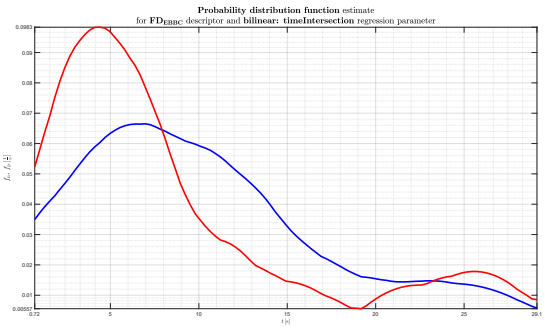
Figure B.2 shows examples of different cases of classes' discernibility illustrated by the estimates of PDFs of their fractal-based tear film surface quality (TFSQ) descriptors for lateral shearing interferometry (LSI) data of healthy vs dry eye syndrome (DES)-affected patients investigated in this thesis.



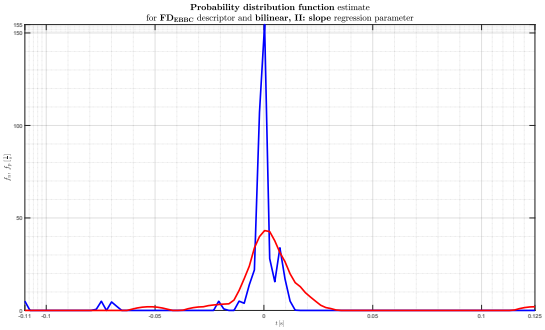
(a) Relatively good discernibility for the null hypothesis that unaffected class mean value is the lower one



(b) Relatively good discernibility for the null hypothesis that unaffected class mean value is the bigger one



(c) Intermediate discernibility



(d) Poor discernibility

Figure B.2: Illustrative cases of PDFs of fractal-based TFSQ descriptors for two classes of patients: unaffected and affected by DES – plotted with blue and red lines, respectively – depicting different levels of discernibility between both groups

C SOURCE CODE LISTINGS

C.1 FRACTAL DIMENSION ESTIMATION

Listing C.1: Box-counting method implementation (chosen parts) (`0e3.1a. DBC vs W-GBC computation times in regard to input image scale - validation and comparison for synthetic fractals and Gaussian noise/src/+fractal/+FD/BC.m`)

```
1  function FDE = BC( img_in, varargin )
2  % BC - Fractal dimension estimation by box-counting.
3  % Basic algorithm.
4  %
5  % https://en.wikipedia.org/wiki/Box_counting
6  %
7  % Example:
8  % FDE = BC(imread('rice.png'));
9  %
10 % Version: 20.07.2019 (Piotr Szyperski).
```

```
16 if nVarargs >= 1 && (ischar(varargin{1}) ||
       isstring(varargin{1})) % first optional argument: 'grayscale'
       (default; synonymous to 'grayscale-p' or 'grayscale-p-image'),
       'grayscale-p-intensity', 'grayscale-w' or 'binary'
17     alg_mode = char(varargin{1}); % choose box-counting variant:
           P-GBC, W-GBC or BBC
18 else
19     alg_mode = 'grayscale';%'binary'; % default value
20 end
```

```
31 if strcmpi(alg_mode, 'grayscale-w') % consider fourth optional
       argument (W-GBC algorithm's resize type) only in case of W-GBC
       algorithm
```

```
32    if nVarargs >= 4 && (ischar(varargin{4}) ||
          isstring(varargin{4})) % fourth optional argument:
          'maximum', 'average' or 'square'
33        w_gbc_alg_submode = char(varargin{4}); % choose W-GBC
              algorithm's resize type (algorithm submode): maximum-,
              average- or square-based
34    else
35        w_gbc_alg_submode = 'maximum';%'average'; % default value
36    end
37 end
```

```
46 if isa(img_in, 'double') % for image classes other than double
       precision
47    img_uncropped = img_in;
48 else
49    img_uncropped = im2double(img_in);%im2double(img_in); %
          convert image to double precision
50 end
```

```
52 if ~ismatrix(img_uncropped) % for RGB color images
53    img_uncropped = rgb2gray(img_uncropped); % convert RGB color
          image to grayscale
54 end
```

```
56 [img, ri, rj] = crop(img_uncropped);
```

Initializations of vectors for box sizes ε_i and numbers of counted boxes N_i

```
58 % Image size:
59 s(1) = size(img, 1);
60 s(2) = size(img, 2);
61 %n_min = 2 + 1;
```

```
62 n_min = 0; % 28.03.2019: left cut-off reduced by three to fit
      algorithm better to Saupe's rescale-and-add-based random
      fractals, as with original cut-off ("n_max = n_gmax - 1;")
      P-GBC-int are declining for Saupe's D values above about 1.65
      causing ambiguity ({FD}_{P-GBC-int}(D) is not an injective
      function)
63 n_gmax = min(nextpow2(mean(s)), 10); % exponent of next greater
      than image size power of 2 (limited to 10 due to slow
      blockproc in maximum-based resize type)
64 %n_max = n_gmax - 1;
65 n_max = n_gmax; % 28.03.2019: right cut-off reduced by one to fit
      algorithm better to Saupe's rescale-and-add-based random
      fractals, as with original cut-off ("n_max = n_gmax - 1;")
      P-GBC-int are declining for Saupe's D values above about 1.65
      causing ambiguity ({FD}_{P-GBC-int}(D) is not an injective
      function)
66 n_vec = (n_min:n_max)';
67 n_n = numel(n_vec);
68 N = zeros(n_n, 1);

70 l_gmax = 2^n_gmax;
71 img = im2double(imresize(im2uint8(img), [l_gmax l_gmax],
      'bilinear'));

73 img = normalize(img);

75 if strcmpi(alg_mode, 'differential')
76    if ~isa(img, 'uint8')
77       img = im2uint8(mat2gray(img));
78    end

80    FDE = fractal.FD.dbc(img);
81 else
82    % Thresholding using Otsu's method:
83    if strcmpi(alg_mode, 'binary')
84       img = imbinarize(img); % image binarisation
85    end
```

Proper internal box-counting procedure

```
87    %% Box-counting
```

```
89    for i = 1:n_n % for all power of 2 box grid divisions from 2^2
          * 2^2 = 4*4 %n = n_min:n_max: n - binary exponent of
          resized image side length
90       n = n_vec(i);
91       if vis
92          fprintf([' Box size %*d of %*d ('...
93             'n = log' char(8322) char(1013) char(8315) char(185)
                ' = %*d, '...
94             char(1013) ' = 2' char(8315) char(8319) ' = %-*.4g,
                '...
95             char(949) ' = ' char(1013) char(183) 'l' char([55349
                56560 8344 8336 8339]) ' = %*d, '...
96             'l = 2' char(8319) ' = %*d'...
97             ').\n'], ...
98             [1+floor(log10(n_n)) i], [1+floor(log10(n_n)) n_n],
                ...
99             [1+floor(log10(n_max)) n], ...
100            [strlength(sprintf('%.4g', 2^-n_max)) 2^-n], ...
101            [strlength(sprintf('%d', 2^(n_gmax-n_min)))
                2^(n_gmax-n)], ...
102            [strlength(sprintf('%d', 2^n_max)) 2^n])
103      end
104      if strcmpi(alg_mode, 'grayscale-w')
105         img_r = resize(img, n, alg_mode, w_gbc_alg_submode);
106      else
107         img_r = resize(img, n, alg_mode);%resize(img, n,
              alg_mode);%resize(img, n, 'maximum');%resize(img,
              n);%resize(img, n, 'average');%resize(img, n,
              'square'); % resize (i.e., scale) the image
108      end
109      N(i) = sum(img_r(:)); % minimal number of boxes
110   end
```

Final fractal dimension estimation implementation

```
117   log2_inv_epsilon = n_vec;
118   log2_N = log2(N);

120   % Linear regression:
121   FDE_reg_par_vec = [log2_inv_epsilon(~isinf(log2_N))
          ones(sum(~isinf(log2_N)), 1)]\log2_N(~isinf(log2_N));
122   FDE = FDE_reg_par_vec(1);
```

Further calculations: for visualization purposes, debugging, performance measurements etc.

```
126     % Logarithm base conversion for plotting purposes:
127     log_inv_epsilon = log10(2)*log2_inv_epsilon;
128     log_N = log10(2)*log2_N;
```

Image resizing algorithm: internal auxiliary procedure of box-counting

```
307 function img_scaled = resize(img_in, n, varargin)

309 %% Optional arguments

311 nVarargs = length(varargin);%nargin-1

313 if nVarargs >= 1 && (ischar(varargin{1}) ||
        isstring(varargin{1})) % first optional argument: 'grayscale'
        (default; synonymous to 'grayscale-p' or 'grayscale-p-image'),
        'grayscale-p-intensity', 'grayscale-w' or 'binary'
314     alg_mode = char(varargin{1}); % choose box-counting variant:
            P-GBC, W-GBC or BBC
315 else
316     alg_mode = 'grayscale';%'binary';
317 end
318 if strcmpi(alg_mode, 'grayscale-w') % consider second optional
        argument (W-GBC algorithm's resize type) only in case of W-GBC
        algorithm
319     if nVarargs >= 2 && (ischar(varargin{2}) ||
            isstring(varargin{2})) % second optional argument:
            'maximum', 'average' or 'square'
320         w_gbc_alg_submode = char(varargin{2}); % choose W-GBC
                algorithm's resize type (algorithm submode): maximum-,
                average- or square-based
321     else
322         w_gbc_alg_submode = 'maximum';%'average'; % default value
323     end
324 end

326 %% Resize algorithm

328 l_0 = size(img_in, 1); % input (original) image side length
```

```
l = 2^n; % resized image side length
s = l_0/l; % image resizing (scaling) factor
%s2 = s^2; % squared image resizing (scaling) factor

if strcmpi(alg_mode, 'grayscale') || strcmpi(alg_mode,
    'grayscale-p') || strcmpi(alg_mode, 'grayscale-p-image') ||
    strcmpi(alg_mode, 'grayscale-p-intensity')
    if strcmpi(alg_mode, 'grayscale') || strcmpi(alg_mode,
        'grayscale-p') || strcmpi(alg_mode, 'grayscale-p-image')
        resize_fun = @(block_struct) ent(block_struct.data(:)); %
            image domain
    elseif strcmpi(alg_mode, 'grayscale-p-intensity')
        resize_fun = @(block_struct) entropy(block_struct.data(:));
            % intensity domain
    end
    img_scaled = blockproc(img_in, [s s],
        resize_fun);%blockproc(img_in_neg, [s s], resize_fun);
elseif strcmpi(alg_mode, 'grayscale-w')
    if strcmpi(w_gbc_alg_submode, 'maximum')
        resize_fun = @(block_struct)
            max(block_struct.data(:));%max(block_struct.data(:)) *
            nnz(block_struct.data) / s2;
        img_scaled = blockproc(img_in, [s s], resize_fun);
    else
        a = repelem(eye(l)/s, 1, s); % intermediate scaling matrix
        if strcmpi(w_gbc_alg_submode, 'average')
            img_scaled = a*img_in*a';
        elseif strcmpi(w_gbc_alg_submode, 'square')
            img_scaled = sqrt(a*img_in.^2*a');
        %elseif strcmpi(alg_mode, 'twenty')
        % img_scaled = (a*img_in.^20*a').^(1/20);
        else
            errorStruct.message = sprintf('Error: unsupported W_GBC
                resize type: ''%s''.\nImage resize type can be either
                ''maximum'', ''average'' or ''square''.',
                w_gbc_alg_submode);
            errorStruct.identifier =
                'resize:unsupportedW_GBCResizeType';
            error(errorStruct)
        end
        %{
        resize_fun = @(block_struct) s2 /
            max(nnz(block_struct.data), 1);
        b = blockproc(img_in, [s s], resize_fun);
        img_scaled = img_scaled.*b;
```

```
      %}
   end
elseif strcmpi(alg_mode, 'binary')
   resize_fun = @(block_struct) any(block_struct.data(:));
   img_scaled = blockproc(img_in, [s s], resize_fun);
end

end
```

Image entropy calculation algorithm: internal auxiliary procedure of box-counting

```
function ent = ent(img_in)

p = img_in/sum(img_in(:));
%ent = nansum(p(:).*log2(1./p(:)));%sum(p(:).*log2(1./p(:)));
ent = -nansum(p(:).*log2(p(:)));%-sum(p(:).*log2(p(:)));

end
```

C.2 P-GBC-IMG ALGORITHM IMPLEMENTATION VALIDATION

Listing C.2: P-GBC-IMG validation on simple matrices (`0c. P-GBC-img validation on simple matrices/src/main.m`)

```
%% Preface

clear;
clc;
%home;
close all hidden

%% Initialization

fprintf(['Program started.' newline])

%% Generate input image matrices
```

```
14 input.generate_input_matrices


17 %% Box-counting

19 % Box-counting procedure parameters:

21 % first optional argument: 'grayscale-p' (synonymous to
       'grayscale'
22 % and 'grayscale-p-image'), 'grayscale-p-intensity,' 'grayscale-w'
23 % or 'binary' (default):
24 % choose box-counting wariant: P-GBC, W-GBC or BBC
25 BC_alg_mode = 'grayscale-p';

27 % second optional argument: false (default value) or true:
28 % turn visualisation off or on
29 %vis = false;

31 % third optional argument (if second optional argument is
       'true'): figure:
32 % number (handle)
33 % set  figure number for visualization
34 %figureNumber = NaN;%1;

36 FDEs = cell(n,1); % FDEs vector - preallocation
37 imgs_p = cell(n,1); % vector of preprocessed images used by
       box-counting algorithm - preallocation
38 imgss_r{i} = cell(n,1); % vector of validation vectors
       (entropies) - preallocation
39 probss = cell(n,1); % probabilities vector - preallocation

41 for i = 1:n
42    img = imgs{i};
43    [ FDEs{i}, imgs_p{i}, imgss_r{i}, probss{i} ] =
          system.BC.BC_validating(img, BC_alg_mode);
44 end

46 %% Results displaying

48 separator = char(ones(1, 50) * char(8212));
49 separator = ['\r\n' separator '\r\n\r\n'];

51 outrelfilename = fullfile('..', 'out', 'out.txt');
52 fileID = fopen(outrelfilename, 'w', 'n', 'UTF-8');%1;
```

```
54 for i = 1:n
55    %img = imgs{i};
56    img_double = imgs_double{i};
57    FDE = FDEs{i};
58    img_p = imgs_p{i};
59    imgs_r = imgss_r{i};
60    probs = probss{i};

62    fprintf(fileID, ['Image %d of %d (%d' char(8201) char(215)
         char(8201) '%d px), BC FDE = %.2f (note that there was no '
         char(949) char(8345) ' sequence cut-offs applied):'
         '\r\n\r\n'], i, n, size(img_double), FDE);

64    %disp(img_double)%disp(img)
65    output.display_mat(img_double, [], fileID);

67    fprintf(fileID, ['\r\n' 'Preprocessed ' char(8211) ' i.a.,
         cropped ' char(8211) ' image used by (internal)
         box-counting algorithm (%d' char(8201) char(215) char(8201)
         '%d px):' '\r\n\r\n'], size(img_p));

69    %disp(img_p)
70    output.display_mat(img_p, [], fileID);

72    fprintf(fileID, ['\r\n' 'Resized sub-images:' '\r\n']);

74    nn = numel(imgs_r);
75    for j = 1:nn
76       jneg = nn-j+1;
77       l_0 = 2^(jneg-1);
78       l = 2^(j-1);

80       fprintf(fileID, ['\r\n' char(9) 'Sub-image %d of %d (grid
            size: %d' char(8201) char(215) char(8201) '%d box; box
            size: %d' char(8201) char(215) char(8201) '%d px) '
            char(8211) '\r\n'...
81          char(9) char(8211) ' counted boxes numbers (i.e., box'
               char(8217) ' entropies: N = -' char(931) char(7522)
               char(8200) 'p' char(7522) char(8201) char(8901)
               char(8201) 'log' char(8322) 'p' char(7522) ')'], j,
               nn, l_0, l_0, l, l);
82       img_r = imgs_r{jneg};
83       img_r(img_r == 0) = 0; % convert negative zeros to zeros in
            img_r for better displaying properties
```

```
84        fprintf(fileID, [', ' char(931) 'N = %.2f:' '\r\n\r\n'],
              sum(img_r(:)));
85        output.display_mat(img_r, char(9), fileID, 1);

87        fprintf(fileID, ['\r\n' char(9) char(9) 'Matrix of p'
              char(7522) ' probability values (box size: %d'
              char(8201) char(215) char(8201) '%d px):' '\r\n\r\n'],
              l, l);
88        prob = probs{jneg};
89        %prob(prob == 0) = 0; % convert negative zeros to zeros in
              prob for better displaying properties
90        output.display_mat(prob, [char(9) char(9)], fileID, l);

92        fprintf(fileID, ['\r\n' char(9) char(9) 'Matrix of log'
              char(8322) 'p' char(7522) ' values (box size: %d'
              char(8201) char(215) char(8201) '%d px):' '\r\n\r\n'],
              l, l);
93        l2p = log2(prob);
94        l2p(l2p == 0) = 0; % convert negative zeros to zeros in l2p
              for better displaying properties
95        output.display_mat(l2p, [char(9) char(9)], fileID, l);

97        fprintf(fileID, ['\r\n' char(9) char(9) 'Matrix of -p'
              char(7522) char(8201) char(8901) char(8201) 'log'
              char(8322) 'p' char(7522) ' values (box size: %d'
              char(8201) char(215) char(8201) '%d px):' '\r\n\r\n'],
              l, l);
98        mpl2p = -prob.*l2p;
99        mpl2p(mpl2p == 0) = 0; % convert negative zeros to zeros in
              mpl2p for better displaying properties

101       output.display_mat(mpl2p, [char(9) char(9)], fileID, l);
102   end

104   if i<n, fprintf(fileID, separator); end
105 end

107 %% Closure

109 if fileID ~= 1 && fileID ~= 2
110   fclose(fileID);
111 end

113 % Display the contents of the saved output file:
114 fprintf(newline)
```

```
fileID = fopen(outrelfilename, 'r', 'n', 'UTF-8');
tline = fgetl(fileID);
while ischar(tline)
   fprintf(['%s' newline], tline);
   tline = fgetl(fileID);
end
fclose(fileID);
fprintf(newline)

%% Epilogue

fprintf(['Program stopped.' newline])
```

C.3 SIMULATION OF FRINGE PATTERN WITH SINUSOIDAL PHASE DISTURBANCE

Listing C.3: Generation of synthetic interferograms with sinusoidal phase disturbance generation (`fd_test/src/fd_test.m`)

```
%#ok<*MSNU>

%% Set flags

calc_noise_fd = false;%true % turn noise fractal dimension
   visualisation on or off (recalculate after image size change)

vis_f = true;%false; % turn exemplary fringes visualisation on or
   off
vis = false;%true; % turn visualisation on or off

%% Set parameters

imsize = 128;%128 % image size

alph = pi/2; % pattern orientation

% Image coordinates:
coord_min = -1;
coord_max = 1;
coord_range = coord_max - coord_min;
```

```
% (Sinusoidal) pattern frequency sweep:
n_f = 100; % number of frequency sweep steps

f_N = (imsize-1)/(2*coord_range);%imsize/2 % Nyquist condition
    frequency

f_min = 0;
f_max = f_N;
f_step = (f_max-f_min)/(n_f-1);

% (Sinusoidal) phase distrubance amplitude sweep:
A_phi_vec = [0 4 15 110 275 500 10000];%[0 1 3 30 89 200
    500];%2.5;%1; % phase disturbance amplitude
n_A_phi = numel(A_phi_vec); % number of phase disturbance angles

n_phi = 2; % number of phase disturbances per phase range

% Pattern orientation disturbance angle sweep:
deg = deg2rad(1);%pi/360;
alph_dist_vec = -5*deg : .5*deg : 5*deg;
n_alph = numel(alph_dist_vec);

%% Variables and vectors initialization

% Pattern frequency/pulsation sweep:
f_vec = f_min:f_step:f_max;
f_rel_vec = f_vec/f_max;
omega_vec = 2*pi*f_vec;

% Phase distrubance frequency/pulsation values:
f_phi = n_phi/coord_range;
omega_phi = 2*pi*f_phi;

fd = zeros(n_A_phi,n_f); % preallocation of fractal dimension
    values

[X,Y] = meshgrid(linspace(coord_min, coord_max, imsize)); % image
    coordinates rectangular grid

%% FD calculation

% Fringe patterns:

if vis
    fig = figure(2); %#ok<UNRCH>
```

```
62    clf(fig,'reset')
63 end

65 for i = 1:n_A_phi % for all phase distrubance frequencies
66    A_phi = A_phi_vec(i);
67    for j = 1:n_f % for all pattern frequencies
68       omega = omega_vec(j);
69       for k = 1:n_alph % for all pattern orientation disturbance
            angles
70          alph_dist = alph_dist_vec(k);
71          alph_val = alph + alph_dist;
72          % Pattern orientation supplementary trigonometrical
               values:
73          sa = sin(alph_val);
74          ca = cos(alph_val);
75          % Time and phase image coordinates rectangular grid
               (i.e., X and Y corrected for pattern orientation)
76          T = X*ca + Y*sa;
77          P = X*sa - Y*ca;

79          Z = sin(omega*T +
               A_phi*sin(omega_phi*P));%cos(0*X).*cos(k*Y)>0;
80          if vis
81             imshow(Z) %#ok<UNRCH>
82             axis xy
83             %maxfig(fig,1);
84             pause(0.04)%waitforbuttonpress
85          end
86          Zbin = Z > 0;
87          fd(i,j) = fd(i,j) + hausDim(Zbin);
88          %fd(i,j)
89          %waitforbuttonpress
90       end
91       fd(i,j) = fd(i,j) / n_alph; % averaging over all pattern
            orientation disturbance angles
92    end
93 end

95 % Noise:

97 if calc_noise_fd
98    avf = 10000; %#ok<UNRCH>
99    fd_noise_temp = 0;
100   for i = 1:avf
101      fd_noise_temp = fd_noise_temp + hausDim(rand(imsize)>0.5);
```

```
   end
   fd_noise = fd_noise_temp/avf;
else
   fd_noise = 1.911104960872848; %#ok<UNRCH> %1.781345203620170;
end

%% Visualisation

font_big = 30;
font_medium = 24;
font_small = 18;

line_width_medium = 2;
line_width_small = 1.2;

n_types = 4;%n_phi/2;

n_colors = n_types;
colors = zeros(n_colors, 3);
gray_max = .85 * ones(1,3);
for i = 1:n_colors % for all phase distrubance frequencies
   colors(i,:) = (i-1)/(n_colors-1) * gray_max;
end
col_num = @(num) mod(num-1,n_colors)+1;

linestyles = {'-';'--'};
n_linestyles = numel(linestyles);
linestyle_num = @(num) min(max(ceil(num/n_types), 1),
   n_linestyles);

fig = figure(1);
clf(fig,'reset')

ax = gca;
%cla(ax,'reset')
set(gca,'XMinorTick','on','YMinorTick','on')
% set(gca, 'TickDir', 'both')

set(ax,'FontSize', font_small)

maxfig(fig,1);
hold on
%plt = plot(f_vec/f_max, fd, 'k', 'LineWidth', line_width);
for i = 1:n_A_phi % for all phase distrubance frequencies
```

```
145     plot(f_rel_vec, fd(i,:), 'LineStyle',
            linestyles{linestyle_num(i)}, 'Color',
            colors(col_num(i),:), 'LineWidth', line_width_small)
146 end

148 lne = line([f_min/f_max f_max/f_max], [fd_noise fd_noise],
        'LineStyle', '--', 'Color', 'k', 'LineWidth',
        line_width_medium);%, 'AlignVertexCenters', 'on');

150 ax.YTick = sort([min(fd(:)) ax.YTick fd_noise max(fd(:))]);
151 %ax.YAxis.TickLabelFormat = '$%,.2f';
152 ax.YTickLabel = cellfun(@(x) num2str(x, '%.3f'),
        num2cell(ax.YTick), 'UniformOutput', false);

154 axis tight

156 grid on
157 grid minor

159 xlabel('$f/f_{N}$', 'Interpreter', 'latex', 'FontSize', font_big)
160 ylabel('$$ \mathrm{FD}_\mathrm{binary} $$', 'Interpreter',
        'latex', 'FontSize', font_big)

162 leg = {'~Fringes, no disr.'; ...
163       '~Fringes, slight disr.'; ...
164       '~Fringes, small disr.'; ...
165       '~Fringes, noticeable disr.'; ...
166       '~Fringes, large disr., intermediate convergence'; ...
167       '~Fringes, large disr., noise convergent'; ...
168       '~Fringes, large disr., extreme case'; ...
169       '~Noise, $p = \frac{1}{2}$'};
170 for i = 1:n_A_phi % for all phase distrubance frequencies
171     A_phi = A_phi_vec(i);
172     leg{i} = [leg{i} ': $A_\varphi = ' num2str(A_phi) '$'];
173 end
174 legend(leg, 'FontSize', font_medium, 'Location', 'best',
        'Interpreter', 'latex')

176 style = hgexport('factorystyle');
177 style.Bounds = 'tight';
178 hgexport(fig, '-clipboard', style, 'applystyle', true);
179 drawnow;

181 %% Exporting figure
```

```
savefig(fig, 'fd_test.fig', 'compact')
print('-dpng', '-r300', 'fd_test')

%% Visualise fringes

if vis_f
   %fig = figure(2); %#ok<UNRCH>
   %clf(fig,'reset')

   % For 17.78 fringes/screen (~14 px fringe wavelength) pattern
      frequency:
   j = floor(n_f/7); %#ok<UNRCH>
   omega = omega_vec(j);

   wavelength = round(2*100/14); % approximate fringe wavelength
      in pixels
   wavelength_str = num2str(wavelength); % string containing
      approximate fringe wavelength in pixels

   filename_prefix = ['fd_test_' wavelength_str 'px_A'];
   ext = '.png';

   % For no pattern orientation disturbance (zero pattern
      orientation disturbance angle):
   alph_dist = 0;
   alph_val = alph + alph_dist;

   % Pattern orientation supplementary trigonometrical values:
   sa = sin(alph_val);
   ca = cos(alph_val);
   % Time and phase image coordinates rectangular grid (i.e., X
      and Y corrected for pattern orientation)
   T = X*ca + Y*sa;
   P = X*sa - Y*ca;

   for i = 1:n_A_phi % for all phase distrubance amplitudes
      A_phi = A_phi_vec(i);

      Z = sin(omega*T +
         A_phi*sin(omega_phi*P));%cos(0*X).*cos(k*Y)>0;
      Zbin = Z > 0;

      %imshow(Z)
      %axis xy
      %maxfig(fig,1);
```

```
223         imwrite(Z, fullfile('Exemplary fringes', [filename_prefix
               num2str(A_phi) ext]), 'ResolutionUnit', 'unknown',
               'XResolution', 300, 'YResolution', 300)
224         imwrite(Zbin, fullfile('Exemplary fringes',
               [filename_prefix num2str(A_phi) '_bin' ext]),
               'ResolutionUnit', 'unknown', 'XResolution', 300,
               'YResolution', 300)

226         %pause(2.50)
227         %waitforbuttonpress
228      end
229   end
```

C.4 GENERATING FRACTAL PATTERN USING RESCALE–AND–ADD APPROACH

Listing C.4: Generation of synthetic fractal images that approximate RFFs based on Saupe's RAA approach (`0e3.1a. DBC vs W-GBC computation times in regard to input image scale - validation and comparison for synthetic fractals and Gaussian noise/src/+fractal/raa2.m`)

```
1  function V_2 = raa2(x, y, varargin)
2  % RAA2 Rescale-and-add method in two dimensions.
3  % Based on:
4  %  “Point Evaluation of Multi-Variable Random Fractals”, Dietmar
      Saupe,
5  % Bremen, Germany, 1989.
6  %
7  % Example:
8  % RF  = raa2(x, y);
9  % RF  = raa2(x, y, D, r, L, Delta, N);
10 % RF  = raa2([1:512].', 1:512, 1.5, sqrt(2), 512, 1, 100);
11 % RF  = raa2(1:512, 1:512, 1.5, sqrt(2));
12 % RF  = raa2([1:512].', 1:512, 1.5, sqrt(2), 512, 1, 100);
13 %
14 % Piotr Szyperski, 14.03.2019.

16 n = 2; % number of dimensions

18 %% Optional arguments
```

```
20 nVarargs = length(varargin);

22 if nVarargs >= 1 && isnumeric(varargin{1}) % first optional
      argument, fractal dimension D: 1 ⩽ D ⩽ 2 (default: 1.5)
23    D = varargin{1}; % choose fractal dimension D
24 else
25    D = 1.5; % default value
26 end
27 D_s = D+1;
28 H = n+1-D_s; % Holder's exponent, 0 ⩽ H ⩽ 1.

30 if nVarargs >= 2 && isnumeric(varargin{2}) % second optional
      argument, lacunarity parameter r: r > 1 (default: √2, often
      used: √2, 2, 4)
31    r = varargin{2}; % choose lacunarity parameter r
32 else
33    r = sqrt(2); % default value
34 end

36 if nVarargs >= 3 && isnumeric(varargin{3}) % third optional
      argument, the largest scale L of the objects: L > 0 (default:
      512, often used: 1)
37    L = varargin{3}; % choose the largest scale L of the objects
38 else
39    L = max(numel(x),numel(y)); % default value
40 end

42 if nVarargs >= 4 && isnumeric(varargin{4}) % fourth optional
      argument, sampling points distance Δ (i.e., resolution 1/N): Δ
       > 0 (default: 1, often used: 1/1000)
43    Delta = varargin{4}; % choose sampling ponits distance Δ
         (i.e., resolution 1/N)
44 else
45    Delta = 1; % default value
46 end

48 if nVarargs >= 5 && isnumeric(varargin{5}) % fifth optional
      argument, number N of random numbers on an edge of a
      n-dimensional hypercube of random numbers to be produced: N ⩾
      1 (default: 100, often used: 100, 50)
49    N = varargin{5}; % choose number N of random numbers on an
         edge of a n-dimensional hypercube of random numbers to be
         produced
50 else
```

```
51      N = 100; % default value
52  end
53
54  %% Parameters
55
56  k_0 = floor(-log2(L)/log2(r)); % -log(L)/log(r); for r=√2, L=512:
        k_0=-18
57  k_max = ceil((1-log2(Delta))/log2(r)); % log(2/Delta)/log(r); for
        r=√2, L=512: k_max=2
58
59  k_1 = k_0 - 2; % for r=√2, L=512: k_1=-20
60  k_2 = k_max + 8; % for r=√2, L=512: k_2=10
61
62  %% Rescale-and-add method
63
64  x = cv(x);
65  y = rv(y);
66  kk = cv(k_1:k_2);
67
68  n_x = numel(x);
69  n_y = numel(y);
70  n_k = k_2 - k_1 + 1;
71  %k = NaN(1, 1, n_k); % preallocation
72  %k = reshape(k_1:k_2, 1, 1, []);
73
74  rmkkH = r.^(-kk.*H);
75  rkk = r.^kk;
76
77  V_2 = zeros(n_x, n_y); % preallocation
78  for i = 1:n_k
79      rmkH = rmkkH(i);
80      rk = rkk(i);
81
82      V_2 = V_2 + rmkH .* S_2(rk.*x, rk.*y, N);
83  end
84
85  %% Debug
86
87  %Z1 = S_2(cv(1:.1:10),rv(1:.1:10)); surf(Z1) % test for S_2
        continuity
88  %Z2 = S_2(cv(0:10:150),rv(0:10:150)); surf(Z2) % test for S_2
        periodicity
89
90  end
```

```
92  function S_2 = S_2(x, y, varargin)
93  % S_2 Piecewise bicubic interpolation.
94
95  %% Optional arguments
96
97  nVarargs = length(varargin);
98
99  if nVarargs >= 1 && isnumeric(varargin{1}) % first optional
        argument, number N of random numbers on an edge of a
        n-dimensional hypercube of random numbers to be produced: N ⩾
        1 (default: 100, often used: 100, 50)
100    N = varargin{1}; % choose number N of random numbers on an
           edge of a n-dimensional hypercube of random numbers to be
           produced
101 else
102    N = 100; % default value
103 end
104
105 %% Initialization
106
107 i = @(x) floor(x);
108 d = @(x) x-i(x);
109 s = @(x) (d(x)).^2 .* (3 - 2*d(x));
110
111 %% Matrix variant
112
113 s_x = s(x);%s(rv(x));
114 s_y = s(y);%s(cv(y));
115 i_x = i(x);
116 i_y = i(y);
117
118 S_2 = ...
119    s_x * s_y .* V(i_x+1, i_y+1, N) + ...
120    (1-s_x) * s_y .* V(i_x, i_y+1, N) + ...
121    s_x * (1-s_y) .* V(i_x+1, i_y, N) + ...
122    (1-s_x) * (1-s_y) .* V(i_x, i_y, N);
123
124 %% Loops variant
125
126 %{
127  %Much slower than matrix variant.
128
129 n_x = numel(x);
130 n_y = numel(y);
131 S_2 = NaN(n_x, n_y); % preallocation
```

```
133 fprintf('Calculating S_2(j,k): %5.1f%%.\n', ((1-1)*n_y + 1-1) /
       (n_x * n_y))

135 for j = 1:n_x
136   x_j = x(j);
137   s_x = s(x_j);
138   i_x = i(x_j);

140   for k = 1:n_y
141     y_k = y(k);
142     s_y = s(y_k);
143     i_y = i(y_k);

145     S_2(j,k) = ...
146       s_x * s_y * V(i_x+1, i_y+1, N) + ...
147       (1-s_x) * s_y * V(i_x, i_y+1, N) + ...
148       s_x * (1-s_y) * V(i_x+1, i_y, N) + ...
149       (1-s_x) * (1-s_y) * V(i_x, i_y, N);

151     fprintf('\b\b\b\b\b\b\b\b\b%5.1f%%.\n', 100*((j-1)*n_y + k) /
          (n_x * n_y))
152   end
153 end
154 %}
155 end

157 function V = V(x, y, varargin)
158 % V Random numbers production.

160 %% Optional arguments

162 nVarargs = length(varargin);

164 if nVarargs >= 1 && isnumeric(varargin{1}) % first optional
       argument, number N of random numbers on an edge of a
       n-dimensional hypercube of random numbers to be produced: N ⩾
       1 (default: 100, often used: 100, 50)
165    N = varargin{1}; % choose number N of random numbers on an
          edge of a n-dimensional hypercube of random numbers to be
          produced
166 else
167    N = 100; % default value
```

```
168     %N = round(sqrt(2)^(ceil((1-log2(1))/log2(sqrt(2))) + 8) *
            512); % default value: (r^k_2) * L =
            r^((1-log2(Delta))/log2(r) + 8) * L = 16384
169 end

171 %% Initialization

173 rng('default')
174 %rng(0)
175 %rng(0, 'twister')
176 %rng(0, 'philox')
177 T = randn(N);%randn(N,N);

179 i = @(x) mod(x-1, N) + 1;
180 V = T(i(x), i(y));
181 end

183 function cv = cv(x)
184 % CV Convert to column vector.

186 cv = reshape(x,[],1);
187 end

189 function rv = rv(x)
190 % RV Convert to row vector.

192 rv = reshape(x,1,[]);
193 end
```

C.5 SEGMENTED LINEAR REGRESSION

Listing C.5: Implementation of bisegmental linear regression with optimization of breakpoints (core part) (`2. Main/src/+piecewiselinfit/lin_biseg_fit.m`)

```
100 % The main loop for the bisegmental linear fit:
101 for indSSE = 1:L2
102     ind = indSSE + 1;
103     x1 = x(1:ind); x2 = x(ind:end);
104     y1 = y(1:ind); y2 = y(ind:end);
105     p1 = polyfit(x1, y1, 1);
106     xi = x(ind); yi = polyval(p1, xi); % the point of intersection
107     % (constraint)
```

```
108     if constrained
109        p2_i = polyfit(x2, y2, 1); % unconstrained LS - to be used
110        % as an initial estimator
111        p2 = fmincon(@piecewiselinfit.bisegmental_error, p2_i, ...
112        [], [], [], [], [], [], ...
113           @piecewiselinfit.bisegmental_const, Options, x2, y2, ...
114           xi, yi); % constrained LS
115     else
116        p2 = polyfit(x2, y2, 1); % unconstrained LS
117     end
118     y1_hat = polyval(p1, x1); y2_hat = polyval(p2, x2);
119     SSE1 = sum((y1-y1_hat).^2); SSE2 = sum((y2-y2_hat).^2);
120     SSE(indSSE) = SSE1 + SSE2;
121  end

123  [~, indSSEi] = min(SSE);
124  indi = indSSEi + 1;

126  x1 = x(1:indi); x2 = x(indi:end);
127  y1 = y(1:indi); y2 = y(indi:end);
128  %y1_avg = nanmean(y1); y2_avg = nanmean(y2);
129  y1_avg = mean(y1); y2_avg = mean(y2);

131  % Second LS fitting - for the set that gave minimum global SSE
132  % (sum of squared errors):

134  p1 = polyfit(x1, y1, 1);

136  xi = x(indi); yi = polyval(p1, xi); % the point of constraint
137  if constrained
138     p2_i = polyfit(x2, y2, 1); % unconstrained LS - to be used
139     % as an initial estimator
140     p2 = fmincon(@piecewiselinfit.bisegmental_error, p2_i, [], ...
141        [], [], [], [], [], @piecewiselinfit.bisegmental_const, ...
142        Options, x2, y2, xi, yi); % constrained LS
143  else
144     p2 = polyfit(x2, y2, 1); % unconstrained LS
145  end

147  y1_hat = polyval(p1, x1); y2_hat = polyval(p2, x2);
```

D | OUTPUT CODE LISTINGS

D.1 VALIDATION OF P-GBC-IMG IMPLEMENTATION ON SIMPLE MATRICES

Listing D.1: Illustrative output code for validation of P-GBC-IMG on image matrix of size $4\,\text{px} \times 4\,\text{px}$ (`0c. P-GBC-img validation on simple matrices_ex_out.txt`)

```
Image 1 of 1 (4 × 4 px), BC FDE = 0.64 (note that there was no ε_n
   sequence cut-offs applied):

    1    0    0    0
    1    1    0    0
    1    1    1    0
    1    1    1    1

Preprocessed - i.a., cropped - image used by (internal)
   box-counting algorithm (4 × 4 px):

    1    0    0    0
    1    1    0    0
    1    1    1    0
    1    1    1    1

Resized sub-images:

  Sub-image 1 of 3 (grid size: 4 × 4 box; box size: 1 × 1 px) -
  - counted boxes numbers (i.e., box' entropies: N =
     -Σ_i p_i · log_2 p_i), ΣN = 0.00:

     0 |   0 |   0 |   0
  -----+-----+-----+----
     0 |   0 |   0 |   0
  -----+-----+-----+----
     0 |   0 |   0 |   0
  -----+-----+-----+----
```

```
   0 |   0 |   0 |   0

  Matrix of p_i probability values (box size: 1 × 1 px):

   1.00 |  NaN |  NaN |  NaN
  ------+------+------+-----
   1.00 | 1.00 |  NaN |  NaN
  ------+------+------+-----
   1.00 | 1.00 | 1.00 |  NaN
  ------+------+------+-----
   1.00 | 1.00 | 1.00 | 1.00

  Matrix of log_2p_i values (box size: 1 × 1 px):

   0.00 |  NaN |  NaN |  NaN
  ------+------+------+-----
   0.00 | 0.00 |  NaN |  NaN
  ------+------+------+-----
   0.00 | 0.00 | 0.00 |  NaN
  ------+------+------+-----
   0.00 | 0.00 | 0.00 | 0.00

  Matrix of -p_i · log_2p_i values (box size: 1 × 1 px):

   0.00 |  NaN |  NaN |  NaN
  ------+------+------+-----
   0.00 | 0.00 |  NaN |  NaN
  ------+------+------+-----
   0.00 | 0.00 | 0.00 |  NaN
  ------+------+------+-----
   0.00 | 0.00 | 0.00 | 0.00

Sub-image 2 of 3 (grid size: 2 × 2 box; box size: 2 × 2 px) -
- counted boxes numbers (i.e., box' entropies: N =
   -∑_i p_i · log_2p_i), ΣN = 5.17:

 1.58 | 0.00
------+-----
 2.00 | 1.58

  Matrix of p_i probability values (box size: 2 × 2 px):

   0.33  0.00 |  NaN   NaN
   0.33  0.33 |  NaN   NaN
  ------------+-----------
```

```
 0.25  0.25 | 0.33  0.00
 0.25  0.25 | 0.33  0.33

Matrix of log2pi values (box size: 2 × 2 px):

-1.58  -Inf |  NaN   NaN
-1.58 -1.58 |  NaN   NaN
------------+-----------
-2.00 -2.00 |-1.58  -Inf
-2.00 -2.00 |-1.58 -1.58

Matrix of -pi · log2pi values (box size: 2 × 2 px):

 0.53   NaN |  NaN   NaN
 0.53  0.53 |  NaN   NaN
------------+-----------
 0.50  0.50 | 0.53   NaN
 0.50  0.50 | 0.53  0.53

Sub-image 3 of 3 (grid size: 1 × 1 box; box size: 4 × 4 px) -
- counted boxes numbers (i.e., box' entropies: N =
   -Σi pi · log2pi), ΣN = 3.32:

+-----+
| 3.32|
+-----+

Matrix of pi probability values (box size: 4 × 4 px):

+-----------------------+
| 0.10  0.00  0.00  0.00|
| 0.10  0.10  0.00  0.00|
| 0.10  0.10  0.10  0.00|
| 0.10  0.10  0.10  0.10|
+-----------------------+

Matrix of log2pi values (box size: 4 × 4 px):

+-----------------------+
|-3.32  -Inf  -Inf  -Inf|
|-3.32 -3.32  -Inf  -Inf|
|-3.32 -3.32 -3.32  -Inf|
|-3.32 -3.32 -3.32 -3.32|
+-----------------------+
```

```
Matrix of -p_i · log_2 p_i values (box size: 4 × 4 px):

+-----------------------+
| 0.33   NaN   NaN   NaN|
| 0.33  0.33   NaN   NaN|
| 0.33  0.33  0.33   NaN|
| 0.33  0.33  0.33  0.33|
+-----------------------+
```

LIST OF TABLES

LIST OF FIGURES

LIST OF CODE LISTINGS

LIST OF ACRONYMS

A

AFM
: atomic force microscopy 21, 154

AUC
: area under the curve 53, 99, 113, 115–117, IX, X, XIII

AVI
: audio video interleave 97, 111, 112, 119

B

BC
: box-counting iii, vii, 11, 15–17, 19, 21, 27–30, 32–35, 39, 40, 47, 50, 51, 57, 58, 61, 82, 83, 85, 86, 105, 118, 124, 129, 134, 139, 141, 143, 151, 152, 155, 156, XV, XVII, XIX, XXI

BBC binary box-counting iii, iv, vii, 19, 27, 33–36, 39, 44, 47–49, 52, 53, 55, 57, 58, 77, 89, 105, 107, 110, 115–120, 122, 123, 125–127, 134, 135, 139, 141, 144, 166

EBBC binary box-counting – variant calculated by the unmodified algorithm, but operating on the binary map of edges detected in the original input image instead of the (binarised) image itself 35, 36, 48, 49, 102, 105, 107, 116–119, 122, 123, 141

DBC differential box-counting iii, vii, 12, 13, 15–19, 27, 37, 50, 53, 55, 57, 82, 85, 87, 106, 107, 116–120, 125, 126, 128, 144, 150, XV

IDBC improved differential box-counting 17, 18, 152

GBC grayscale box-counting v, viii, 37, 82, 83, 85, 86, 106, 141

P-GBC probabilistic grayscale box-counting 37, 38, 47, 49, 61–63, 68, 72, 82, 83, 141

P-GBC-IMG probabilistic grayscale box-counting – variant based on entropy using probabilities determined in *image* (spatial) probability space iii, vii, 37, 50, 52, 53, 55, 57, 61, 72, 74, 82, 85, 87, 120, 125–127, 141, 143, XXI, XXXIX

P-GBC-IMG-GLB probabilistic grayscale box-counting based on image probability space – globally calculated variant 61, 62, 73, 74, 127, 134, 135, 141

DDR
: double data rate

DDR3 double data rate type three 113

DES
: dry eye syndrome iv, viii, 52, 53, 56–58, 96–99, 104, 106, 108, 113, 114, 116, 117, 122, 143, 144, XIII, XIV

F

FAZ
: foveal avascular zone 21

FD
: fractal dimension iii–v, vii, viii, 9–21, 23, 25, 27, 28, 30, 36, 37, 39, 47, 51–53, 55, 56, 58, 59, 61, 77, 78, 82–85, 87, 89, 91, 105–107, 109–111, 115–119, 122–129, 132, 134, 135, 139, 143–146, 149–154, 156, VII, XV, 166

FDE fractal dimension estimation iii, iv, vii, viii, 3, 9–14, 17, 19, 20, 23, 27, 35, 36, 43, 50, 51, 54, 61, 72, 75, 77, 89, 93, 95, 101, 105, 109, 118, 123–125, 127, 129–134, 139, 141, 143, 144, 147, 151, 152, 155, III, IV, XV, XVIII

FNR
: false negatives rate IX

FPR
: false positives rate IX

FT
: Fourier's transformation

FFT fast Fourier's transformation 12, 105, 160

fBf
: fractional Brownian function 12, 15, 18, 19, 152

fBm
: fractional Brownian motion 14, 18, 151, 152

I

IDE
: integrated development environment 81

IEEE
: Institute of Electrical and Electronics Engineers 27, 124, 147, 149–153, 155, 158–160

IFS
: iterated function system 19

infoGain
: information gain quality function IX

K

KDE
: kernel density estimation 99, 113, X–XII

L

LoG
: Laplacian of Gaussian 105

LSI
: lateral shearing interferometry iv, viii, 35, 52, 57, 77, 82, 89, 90, 95–98, 100–102, 104, 106, 109, 111, 119, 140, 141, 144, 148, 158, 160, III, XIII

M

Matlab
: matrix laboratory – a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks 19, 50, 81, 100, 102, 103, 106, 113, 120, 122, 125, 129, 152, 159

MS
: Microsoft Corporation 113

MSE
: mean squared error 13, 14, 16, III

MISE mean integrated squared error XII

AMISE asymptotic mean integrated squared error 99, XI

MVUE
: minimum-variance unbiased estimator 124

O

OCT
: optical coherence tomography 154

OCTA optical coherence tomography angiography 21, 154

P

PC
: personal computer 113

PDF
: probability density function XII–XIV

PNG
: portable network graphics 77, 125

R

RAA
: rescale-and-add iv, viii, 77, 78, 82, 84, 85, 88, 119–121, 140, XXXI

RAM
: random-access memory 113

DRAM dynamic random-access memory

SDRAM synchronous dynamic random-access memory 113

RCC
: reticular cell counting 13–16, 19

PRCC probabilistic reticular cell counting 14, 16

IPRCC improved probabilistic reticular cell counting 14, 16

RFF
: random fractal function iv, viii, 77, 78, 81, 82, 84, 85, 140, XXXI

ROC
: receiver operating characteristic iv, viii, 53, 57, 99, 113, 115–118, 161, IX, X, XII, XIII

S

SBC
: suppressed blinking condition 96

SI
: *Système international d'unités* 37

SSD
: solid-state drive 113

T

TFSQ
: tear film surface quality 20, 89, 90, 95, 99, 105–109, 111, 113–119, 121–123, 147, 148, 156, 158, 160, XIII, XIV, 166

TNR
: true negatives rate IX

TPR

true positives rate IX

W

WRAcc

weighted relative accuracy (novelty) quality function IX