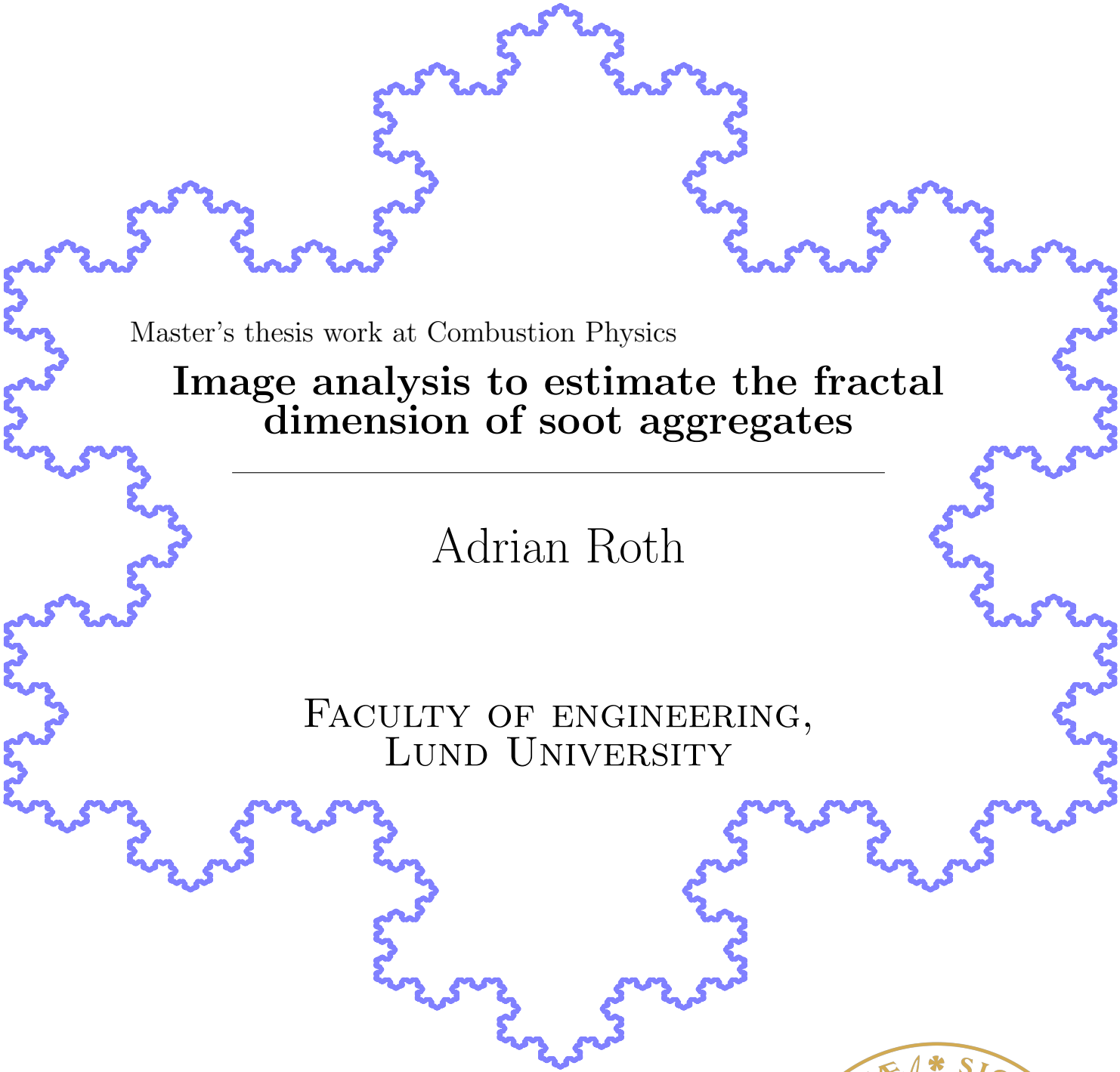

Master's thesis work at Combustion Physics

Image analysis to estimate the fractal dimension of soot aggregates

Adrian Roth

Faculty of engineering,
Lund University

Supervisors: Per-Erik Bengtsson & Kalle Åström
Date: 2018 - 02 - 26

Abstract

In this master's thesis a method for estimating the fractal dimension from Transmission Electron Microscope (TEM) images of soot aggregates has been developed, and also applied to TEM images of soot sampled from a mini-CAST soot generator. The problem to estimate the fractal dimension, a 3D property, from a 2D image is undetermined. To solve this, TEM images were synthesised based on projections of numerically calculated soot aggregates with specified fractal dimensions. The images were used to calibrate a machine learning approach with the output of a fractal image analysis method, called the Box Counting dimension, as fractal feature. The created calibrations show a strong correlation between the Box Counting dimension and the fractal dimension. This verify that the Box Counting dimension as a feature extracts the fractal properties from a TEM image. Two different versions of Box Counting, named standard-Box Counting and random-Box Counting with the difference in how the boxes are generated, have been tested and compared. The random-Box Counting method was found to give the most reliable calibration and estimations. This calibration is similar to a corresponding one produced with the same method in previous work. 55 TEM images of soot sampled from the cold gas exhaust of a mini-CAST soot generator were analysed. A preprocessing algorithm was applied on each image followed by an estimation of the fractal dimension where the calibrations were used. The resulting mean of all fractal dimension estimations from the random-Box Counting method was 1.94. The evaluated fractal dimensions in this work is somewhat higher compared to studies probing flame soot and possible reasons are discussed.

Examensarbetet har utvecklat en metod för att estimera den fraktala dimensionen från Transmissions Elektron Mikroskop (TEM) bilder av sot aggregat som skapats med hjälp av en mini-CAST sotgenerator. Det är ett underbestämt problem att estimera den fraktala dimensionen, en 3D egenskap, från 2D bilder. Lösningen är syntetiserade TEM bilder som baserats på projektioner av numeriskt simulerade aggregat med specifika fraktala dimensioner. Bilderna användes för att kalibrera en maskininlärningsmetod med värdet från en fraktal bildanalysmetod, kallad Box Counting dimensionen, som fraktalfeature. De skapade kalibreringarna visar en stark korrelation mellan Box Counting dimensionen och den fraktala dimensionen. Detta verifierar att Box Counting dimensionen, som feature, extraherar de fraktala egenskaperna från en TEM bild. Två olika versioner av Box Counting, kallade standard-Box Counting och random-Box Counting med skillnad i hur boxarna genereras, har testats och jämförts. Random-Box Counting metoden visade sig ge den mest pålitliga kalibreringen och de mest pålitliga estimeringarna. Denna kalibrering är lik en motsvarande kalibrering som skapats med hjälp av samma metod i ett tidigare arbete. 55 TEM bilder av sot som samplats från den kalla avgasluften av en mini-CAST sotgenerator har analyserats. En förbehandling av varje bild följdes av en estimering av den fraktala dimensionen där kalibreringarna från maskininlärningen användes. Det resulterande medelvärdet av alla estimerade fraktala dimensioner från random-Box Counting metoden var 1.94. De evaluerade fraktala dimensionerna i detta arbete är lite högre jämfört med studier som samplat sot direkt i flammor och möjliga orsaker diskuteras.

Acknowledgement

This master's thesis would not have been possible without the support from my supervisors, Per-Erik Bengtsson and Kalle Åström, friends and family especially, Anton Roth and Sara Olsson. In all of my most exciting and doubting moments you have always found time for discussion, creativating ideas or simply being there for me. I am most grateful for this, thank you!

Contents

Chapter 1

Introduction

Since the dawn of time humans have searched for patterns. From finding constellations in the stars to the search for Romanescos in the clouds. Naturally it can be expanded to finding patterns inside patterns. Here the phenomenon of self-similarity might arise. For example, zooming in on one sub cone of a Romanesco cauliflower, see Figure 1.2, will almost repeat the exact same pattern as the one seen in the whole vegetable. Self-similarity is connected to the world of fractal geometry, which is a concept introduced by the mathematician Benoit B. Mandelbrot. His legacy includes work about fractals in nature [1] which is exactly what this master's thesis will explore. Here we will be zooming in about 100 000 times into the small world of soot aggregates and their fractal geometry.

First consider, what is soot? A soot particle is a solid substance mainly consisting of carbon and to less extent hydrogen [2]. Soot aggregates or soot clusters are clusters of these particles. The creation of soot aggregates starts with the collision of particles. Larger aggregates are then mostly created through cluster-cluster collision. Since all aggregates, independent of size, are built up of sub-aggregates created from the same process they will possess self-similar properties. This also means that soot aggregates have fractal properties.

In urban areas soot is mainly produced from the combustion of fossil fuels. The global soot emissions are one of the main causing factors for the global warming [3]. The presence of soot in the atmosphere contribute to so-called climate forcing through absorption of the solar radiation both on the ground and in the air. In the end this leads to an increase of the earth surface temperature. In addition to the environmental viewpoint, soot emissions are strongly connected to health risks [4]. Meta-analyses indicate that the fraction of people affected by respiratory and cardiovascular diseases increase for people living in areas with high particulate air pollution.

The motivation for the present study on the properties of soot is connected to two currently ongoing connected projects at Lund University. The first one, at Combustion Physics, is a study of the optical properties of soot by Phd. S. Török and Prof. P.-E. Bengtsson. In this project soot is produced with a so-called mini-CAST soot generator. The generator simply produces soot from combustion of propane gas. The produced soot can be tuned to have different properties from nascent small relatively transparent soot to mature larger strongly absorbing soot. The focus of the project is on measuring the absorption properties together with scattering properties of soot. Results show interesting differences in scattering properties for the tuned soot aggregates which are related to their fractal properties. The second project is theoretical work done by Prof. A. Karlsson, Electro and Information Technology, in collaboration with P.-E. Bengtsson. By the means of theoretical calculations and synthesised aggregates the optical properties of soot are studied. One input parameter for computing the synthesised aggregates is a fractal parameter called the fractal dimension. The results so far show a critical correlation between the fractal dimension of a synthesised aggregate and its scattering properties.

This master's thesis aims to act as a bridge between these two projects. Both projects investigate the optical properties of soot aggregates but in different realms, the real and simulated. If soot aggregates with the same fractal dimension are studied in both projects, the agreement between measurements and calculations would be improved. In the end this will increase the knowledge of,

and improve, the current models for the optical properties of soot aggregates.

To act as a bridge the work presented in this master's thesis needs to estimate the fractal dimension of soot aggregates. Here two questions follow: What is the fractal dimension and how can it be estimated for soot aggregates?

The fractal dimension is a property not that different from the topological dimension where a curve has the topological dimension 1, a surface 2 and volume 3. The difference is the possibility for non integer fractal dimensions. All possible fractal dimensions include the continuous interval of real values between 1 and 3. As a short example explaining the fractal dimension, a straight line will both have the topological and fractal dimension 1. Though as the line evolves into a curve the fractal dimension increases. Adding more turns, especially sharp ones, on the curve will increase it further. It should be noted that the fractal dimension is not the most intuitive concept, a more elaborate discussion on it is found in the theory.

To calculate the fractal dimension value, one way is too use image analysis on Transmission Electron Microscope (TEM) images of soot aggregates, just like the one in Figure 1.3. But this creates a problem. The fractal dimension of a soot aggregate is a 3D property and trying to estimate it from a 2D image is an undetermined problem, meaning it is impossible to know the correct answer. In this project the problem of not knowing the correct answer is solved with the use of synthesised TEM images. These are projected from soot clusters simulated with tunable algorithms for generation of fractal like aggregates, which is exactly what Prof. A. Karlsson use in his project introduced above. Now the correct fractal dimension for each TEM image is known. The images are used to calibrate a machine learning approach where the output of a fractal image analysis method, called the Box Counting dimension, is used as the fractal feature. Machine learning is a general term connected to using data driven learning rather than explicit programming. The method can thereby learn from the input and output in previous data to predict the output of new input data. This approach of synthesising TEM images and calibrating a machine learning method for estimating the fractal dimension of soot aggregates has been applied in the work of Wozniak et al. [5]. This master's thesis is mostly a reproduction of their method. The differences between the projects lie in using different simulation and projection methods for synthesising TEM images and an improved description of an automated preprocessing algorithm which can be applied on TEM images. The methods used in this report are generally faster versions.

The aim of the thesis is two-fold:

- To develop an accurate image analysis method for estimating the fractal dimension from TEM images of soot aggregates using synthesised TEM images.
- Applying the developed method on real TEM images of soot aggregates.

To estimate the fractal dimension this work will, as mentioned, use a machine learning approach. Machine learning is a process applying the concept, the more data the better, where the data in this project will be synthesised TEM images. For it to be practical, with the aspect of time, to apply the pipeline of synthesising images and calibration. Much focus and work has been put into fast implementations of the used methods.

Behind the visualisation of all results of this project, lie more than 4000 lines of code. These are a mix of the languages C, C++, Python, Matlab, BASH and Julia which together form a flow of data, creating files, images and plots illustrated in the flowchart in Figure 1.1.

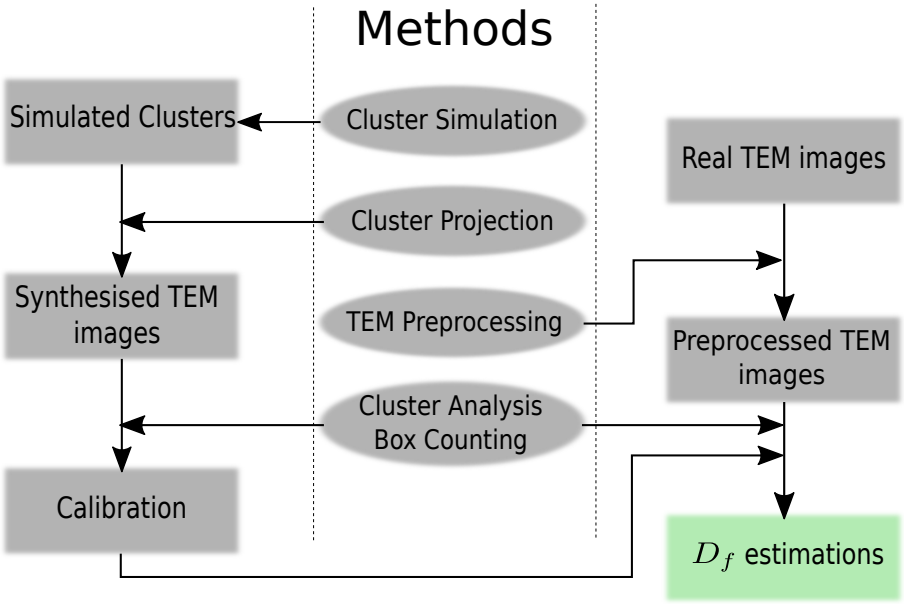

Figure 1.1: Flowchart of all implemented methods and how they interact with the data in this project where D_f is the final fractal dimension estimations of the real TEM images. The ellipses represent implemented methods and the rectangles are data produced with these methods.

The report will continue with a chapter containing theory on producing the synthesised TEM images, image analysis methods for preprocessing images and retrieving fractal parameters. This is followed by the results and discussion of images and plots mainly produced by the calibration and fractal dimension estimation of real TEM images. The final chapter contains the conclusions of the discussion and how they relate to the aims introduced above.

Figure 1.2: A Romanesco Cauliflower with clear self-similarity at different scales and fractal properties.

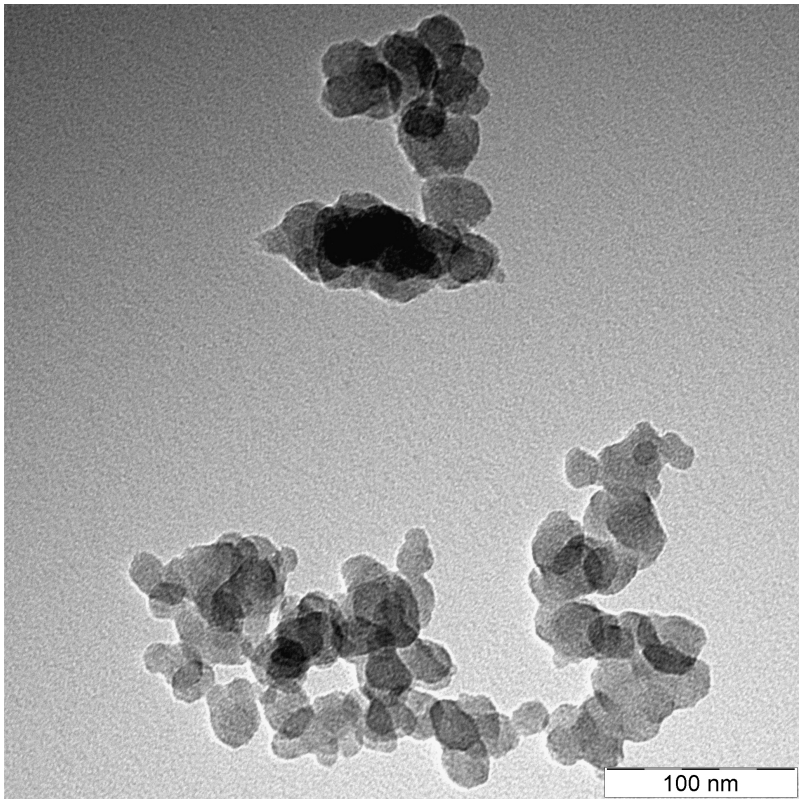

Figure 1.3: Example of a Transmission Electron Microscope image with two soot aggregates produced from a soot generator. Note that the soot aggregates consists of primary soot particles, which are roughly spherical and have diameters around 20 nm.

Chapter 2

Theory

2.1 Fractals

The defining properties of fractals are revealed when observing them at different scales. Here the pattern in each scale will be similar which is called self-similarity. For example, theoretical fractals often have perfect self-similarity. Scaling them will repeat the exact same pattern over and over. One theoretical fractal, with perfect self-similarity, called the Koch snowflake is illustrated in Figure 2.1.

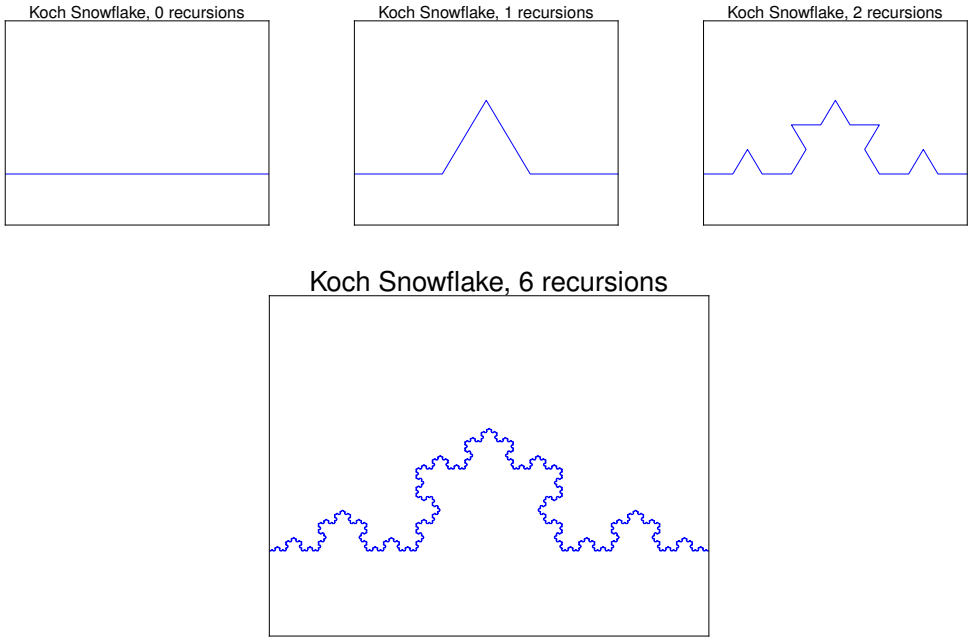

Figure 2.1: This is an illustration of the Koch snowflake, a theoretical fractal with perfect self-similarity at different scales. The top row of images show how the fractal is produced. In each recursion step all lines are divided into four new lines with a third of its original length. This is repeated indefinitely. Six such recursions are shown in the last image.

To connect self-similarity and scaling to the fractal dimension it is interesting to look at the fractal scaling equation:

$$N_L \sim L^{-D_f}. \tag{2.1}$$

For fractal scaling the relation above should hold where N_L is the number of sticks with length L needed to measure the length of an object with the fractal dimension D_f. Solving the fractal

scaling equation for D_f with two N_L, L combinations gives:

$$\begin{aligned} N_L &\sim L^{-D_f}, \\ \log N_L &\sim -D_f \log L, \\ D_f &= -\frac{\Delta \log N_L}{\Delta \log L}. \end{aligned} \tag{2.2}$$

For the fractal scaling equation to make any sense it will be applied to the snowflake above as an example. Here the length of the fractal will be measured using two sticks with different lengths. The first one will have the length $L_1 = 1$ and the second one $L_2 = \frac{1}{3}$. When measuring the straight line in the top leftmost image in Figure 2.1 with the first stick, it fits perfectly, meaning that one stick was needed to measure its length, $N_1 = 1$. Now using the second stick the *resolution* is suddenly improved to measure the same line in the top middle image of Figure 2.1 which is divided into four lines where each line has a third of the previous line's length. Conveniently the new stick also has a third of the previous stick's length and therefore four sticks are needed to measure the length, $N_2 = 4$. Inserting the values into the resulting expression for D_f in Eq. (2.2) gives:

$$D_f = -\frac{\log(N_2) - \log(N_1)}{\log(L_2) - \log(L_1)} = 1.26. \tag{2.3}$$

So each time the scale is doubled the length of the snowflake curve will be $2^{1.26} \approx 2.39$ times larger. The Koch snowflake is not just a set of lines with one dimension, but something more, something with fractal dimension 1.26.

The next equation is connected to both the fractal scaling equation and aggregates. It is called the fractal equation [5], note the similarity of names, and is shown below.

$$n_p = k_f \left(\frac{R_G}{r_p} \right)^{D_f}. \tag{2.4}$$

Here n_p is the number of particles in the aggregate, k_f is called the fractal prefactor which among other things is a measure of how much the particles in the cluster overlap in relation to hard spheres in contact, R_G is the radius of gyration for the aggregate, r_p is the radius of each particle and D_f is as you might have guessed the fractal dimension. The fractal equation is widely used to describe the fractal properties of aggregates, including soot aggregates. For soot aggregates it is used in a certain method to estimate the fractal dimension from TEM images.

Relating the fractal dimension to real soot aggregates Table 2.1 shows results of fractal dimension estimations in some previous works.

Table 2.1: Estimated values of the fractal dimension for soot aggregates from previous works. The soot aggregates used for these calculations have been sampled in a flame burning with either diesel or ethene. The methods mentioned here will be explained later in section 2.4.

Wozniak et al. [5], using the Minimum Bounding Rectangle method	1.88
Wozniak et al. [5], using the random-Box Counting method	1.66
Dobbins and Megaridis [6], using the Minimum Bounding Rectangle method	1.62 & 1.74

2.2 Cluster Simulation of aggregates with predefined fractal dimension

The simulation of fractal-like clusters is a problem present in multiple science-fields. There are a number of different methods to perform this, e.g. in Wozniak et al. [5], Thouy and Jullien [7] and Skorupski et al. [8]. In this master's thesis two different simulation methods have been used. The first one is based on the method found in Thouy and Jullien [7]. This is a Cluster-Cluster (CC)

model, meaning that in each step of the algorithm two clusters of particles are merged. The second model is a Particle-Cluster (PC) method described in Skorupski et al. [8]. Here each step of the algorithm adds a single particle to the current aggregate. Both methods require the input of a specific number of particles (n_p) and a fractal dimension ($D_f \in [1, 3]$) to produce a corresponding aggregate.

2.2.1 Cluster-Cluster aggregation model in unit grid

This method is as mentioned based on the description by Thouy and Jullien [7]. In their article a more thorough deduction of formulas are shown which will not be included here. The algorithm simplifies the clusters into an integer lattice, meaning that the coordinates for each particle in a cluster can only take integer values. The method is also restricted to only produce clusters with $n_p = 2^n, n \in \mathbb{N}$ particles. For notation, a cluster of particles can be described using a list of point-coordinates for each particle in the cluster. The points are in 3D and will be denoted $\mathbf{r}_i$ for the i'th particle in the aggregate.

Now to produce an aggregate with 2^n particles, first 2^{n-1} clusters with two particles in each are created. The creation is performed randomly meaning that the second particle is randomly inserted into one of the three possible directions relative the first one. It should be noted that this is the only part of the algorithm containing a random component. Then as mentioned the clusters should be merged together and the algorithm only describes how this is done for two general clusters with equal number of particles and the rest should be performed recursively. Each recursive step uses the clusters produced in the last step merging them together, resulting in half as many clusters until there is only one cluster left.

But to understand the algorithm it is good to have some formulas fresh in memory. One way to describe the size of a cluster is using the Radius of Gyration (R_G). This is calculated as

$$R_G = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |\mathbf{r}_i - \mathbf{r}_G|^2}, \tag{2.5}$$

where $\mathbf{r}_G$ is the centre of gravity of the cluster. If two clusters with equal number of particles are merged the radius of gyration of the merged cluster (R_G) can be calculated from the radius of gyration of the previous clusters (R_{G1} and R_{G2}) together with their respective centre of gravities ($\mathbf{r}_{G1}$ and $\mathbf{r}_{G2}$) as

$$R_G^2 = \frac{(R_{G1}^2 + R_{G2}^2)}{2} + \frac{|\mathbf{\Gamma}|^2}{4}, \tag{2.6}$$

where

$$\mathbf{\Gamma} = \mathbf{r}_{G1} - \mathbf{r}_{G2}. \tag{2.7}$$

As mentioned, the created clusters should have a predefined fractal dimension. This can be achieved through preserving the fractal dimension of the clusters in each merging. With the assumption that the two clusters about to be merged have the equal known fractal dimension D_f. The new problem is to preserve this fractal dimension after the merging. The solution will use a reformulation of the fractal equation (2.4):

$$n_p \sim R_{G,n_p}^{D_f}. \tag{2.8}$$

To both preserve D_f and double the number of particles when merging two clusters a restriction is set on how the radius of gyration for the new aggregate can change. The only flexible part in the new radius of gyration is $|\mathbf{\Gamma}|^2$. From Eq. (2.6) it can be seen that $|\mathbf{\Gamma}|^2$ should be described by the following expression:

$$|\mathbf{\Gamma}|^2 = k^2 \frac{R_{G1}^2 + R_{G2}^2}{2}, \tag{2.9}$$

where the fractal dimension is a component of k as:

$$k = 2\sqrt{4^{1/D_f} - 1}. \tag{2.10}$$

Now in practice the following equation will be used instead of (2.9):

$$|\mathbf{\Gamma}|^2 = k^2 \frac{R_{G1}^2 + R_{G2}^2}{2} + 1. \tag{2.11}$$

The 1 here is introduced for the formula to work for the base case with two $N = 1$ clusters and for larger clusters it will have insignificant effect.

Now to the real algorithm of merging two general clusters with equal number of particles while preserving the fractal dimension. First all possible ways of combining the clusters, excluding those with particles overlapping, are found. Since particles only exist in an integer lattice this is a finite number of combinations. For each combination a value ($\delta\Gamma$) describing how good it preserves the fractal dimension, where smaller is better, is calculated as:

$$\delta\Gamma = \left(|\mathbf{\Gamma}|^2 - k^2 \frac{R_{G1}^2 + R_{G2}^2}{2} - 1 \right)^2. \tag{2.12}$$

The combination with the smallest value is chosen as the merged cluster. To explain the appearance of $\delta\Gamma$ it might be easier to look at the previous representation of $|\mathbf{\Gamma}|^2$ in (2.11) which describes its optimal value for preserving the fractal dimension. Now it is more intuitive that the best cluster is the one minimising the difference between the real $|\mathbf{\Gamma}|^2$ for a certain combination of the two clusters and the optimal one.

Most of the cons for this method such as the restrictions of a lattice grid, slow computation and the restricted number of particles in an aggregate has been mentioned. Though there is one advantage. Cluster-Cluster aggregation is a realistic way of producing aggregates since it follows the theory of the production of real soot aggregates.

2.2.2 Particle-Cluster aggregation model in a continuous space

The second model is as mentioned described in Skorupski et al. [8]. It adds a single particle to the cluster in each aggregation step while still preserving the fractal dimension. To preserve the fractal dimension the new particle should be added at a certain radius from the cluster's centre of gravity. This radius $|\mathbf{\Gamma}|^2$ is described by [8]:

$$|\mathbf{\Gamma}|^2 = \frac{n_p r_p^2}{n_p - 1} \left(\frac{n_p}{k_f} \right)^{\frac{2}{D_f}} - \frac{n_p r_p^2}{n_p - 1} - n_p r_p^2 \left(\frac{n_p - 1}{k_f} \right)^{\frac{2}{D_f}}. \tag{2.13}$$

Here n_p is the number of particles in the cluster *after* the particle is added, r_p the radius of each particle, k_f the fractal prefactor which is set to 1.593 just as in Wozniak et al. [5] and D_f the preset fractal dimension of the resulting cluster. But the first thing to do in this method is to create a two particle cluster at random. It can be performed by taking two uniform random numbers, $\hat{\theta} \in [0, \pi]$ and $\hat{\phi} \in [0, 2\pi)$. The second particle is set, with spherical coordinates, to $(2r_p, \hat{\theta}, \hat{\phi})$ with the first particle as origin. Thereafter the same following procedure is repeated for adding a single particle (A) to the cluster until the desired number of particles is reached.

1. The radius $|\mathbf{\Gamma}|$ for the new particle A from the cluster's centre of gravity C is calculated using Eq. (2.13).

2. The particles in the cluster with radius r matching the condition $|r - |\mathbf{\Gamma}|| < 2r_p$ are extracted. These are particles on which A can be attached while still fulfilling the radius condition.

3. One of these particles B with radius r_B from C is chosen at random.

4. For the radius condition to be met the angle $\angle ACB$ is restricted to one value α. This angle can be calculated using Eq. (A.1) with $|\mathbf{CA}| = |\mathbf{\Gamma}|$, $|\mathbf{CB}| = r_B$ and $|\mathbf{AB}| = 2r_p$.

5. There is now one degree of freedom left for A to be attached to B. The freedom is the angle $\beta \in [0, 2\pi)$ of the vector $\mathbf{CA}$'s rotation around $\mathbf{CB}$. The angle can be extracted from a uniform distribution and finding the cartesian coordinates for A is performed using the rotations described in Appendix B.

6. Check if A is colliding with any other particles in the cluster.

   if yes for the 25th time [8]: goto 3
   if yes: goto 5
   if no: add particle A to cluster

In addition to this procedure two problems were encountered which were not accounted for in the algorithm described by Skorupski et al. [8]. Here special cases have been adapted.

- If the radius of the new particle is larger than the radius of the particle farthest away from the centre. The solution is to insert the new particle on this particle as far away from the cluster centre as possible. Occurs for D_f values close to 1.
- If the radius is too small. Meaning that all of a large number of attempted insertions of the new particle has resulted in collisions with previous particles. The solution is to increment the radius and try again. Occurs for D_f values close to 3.

The pros and cons for the PC simulation method follows the same pattern as for the CC method, but inverted. Here the advantage is a fast algorithm with continuous coordinates for an arbitrary number of particles in the clusters. The drawback is an unrealistic way of producing aggregates by adding one particle at a time and for large preset fractal dimensions ($D_f > 2.9$) the algorithm has a tendency to get slow because of the second special case above.

2.3 Cluster Projection, synthesising TEM images from simulated clusters

The principle of a Transmission Electron Microscope is mostly the same as a regular microscope [9] with the difference that a TEM uses electrons and the regular microscope photons. The main advantage of a TEM is that it has much higher resolution. To such degree that macromolecules, such as soot aggregates, can be viewed.

The projection of simulated aggregates should imitate the operation of a TEM. One approach to this problem of projecting a 2D image from 3D points is connected to the camera matrix. This is a matrix with three rows and four columns. The general camera matrix ($\mathbf{C}$) includes information of the camera centre, its rotation, the focal length, etc. With the camera matrix a 3D point ($\mathbf{X}$) as column vector, which could be the centre of a simulated particle, can easily be projected into a pixel value ($\mathbf{x}$) from simple matrix multiplication:

$$\mathbf{x} \sim \mathbf{C}\mathbf{X}. \tag{2.14}$$

To be able to compute the multiplication $\mathbf{X}$ must be a 4 long column vector and $\mathbf{x}$ a 3 long column vector. This is called homogeneous coordinates where a third or fourth homogeneous coordinate is added in two and three dimensions respectively. Converting from normal to homogeneous coordinates is simple. Just add the homogeneous coordinate as a 1. To convert the homogeneous coordinate back, first scale the whole vector so that the homogeneous coordinate is 1. Then it can be removed. This explains the proportionality in Eq. (2.14) where $\mathbf{x}$ should be scaled to get the homogeneous coordinate as 1.

The camera matrix can be simplified into an intrinsic- ($\mathbf{K}$ 3x3), rotation- ($\mathbf{R}$ 3x3) and translation- ($\mathbf{t}$ 3x1) matrix on the form:

$$\mathbf{C} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]. \tag{2.15}$$

The intrinsic matrix contains information of the camera such as focal length (f), conversion from measuring units to pixels (s_x and s_y), skew (γ) and the centre of an image in pixels (u_0, v_0). A model for the intrinsic matrix is the following:

$$\mathbf{K} = \begin{bmatrix} \frac{f}{s_x} & \gamma & u_0 \\ 0 & \frac{f}{s_y} & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.16}$$

The rotation matrix is connected to the rotation of the camera's coordinate system. It describes the inverse rotation from the cartesian coordinate vectors to the camera's coordinates. For example, the camera's direction vector ($\mathbf{v}$) be calculated using the z unit vector as:

$$\mathbf{v} = \mathbf{R}^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{2.17}$$

At last the translation matrix (column vector) does *not* describe the real camera centre. Instead the real camera centre ($\mathbf{T}$) can be calculated as $\mathbf{T} = -\mathbf{R}^T\mathbf{t}$. $\mathbf{t}$ somehow describes the camera centre after all points has been rotated by $\mathbf{R}$.

A special case for the camera, called an affine camera, is when the camera centre is set at infinity. This is an interesting approximation when observing objects far away or small object where large magnification has been used. To apply this, the camera matrix in Eq. (2.15) is modified to:

$$\mathbf{C} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.18}$$

where $\mathbf{R}$ and $\mathbf{t}$ now only has two rows. For further reading about the camera matrix and computer vision problems see Hartley and Zisserman [10].

Wozniak et al. [5] use a 3D overlap between particles where they try to mimic the morphology of real soot aggregates while projecting them. This is described using an overlapping factor C^{3D} described below:

$$C^{3D} = \frac{2r_p - d^{3D}}{2r_p}, \tag{2.19}$$

where d^{3D} is the real distance between the centres of two particles and r_p is the radius of each particle. They used the overlapping factor, $C^{3D} = 0.2$ for synthesising TEM images which is estimated from the corresponding overlap in real TEM images of soot aggregates.

2.4 Cluster Analysis, methods for estimating the fractal dimension from TEM images

Just like for the simulation, two methods are described here, with the addition of a small variation of the second one. They are explained in Wozniak et al. [5] with the names: Minimum Bounding Rectangle (MBR) method and Box Counting method, with the variation as the Modified Box Counting method. The first one is based on the fractal equation (2.4) while the others are based on the fractal scaling equation (2.1).

2.4.1 Minimum Bounding Rectangle

Even though this is a method widely used it has no specific name in e.g. Köylü et al. [11] and Dobbins and Megaridis [6]. But Wozniak et al. [5] decided to name it after the way it needs the input images to be rotated and cropped. This should be performed in a way as to minimise the area of the resulting image, in other words finding the minimum bounding rectangle. The method then has its base in two estimations. The first one is an estimation of the radius of gyration as the characteristic length of the minimum bounding rectangle. In an image with pixels height/length as L and the width W of this rectangle the characteristic length is $\sqrt{LW}$. Now the fractal equation (2.4) can be rewritten as:

$$n_p = k_{LW} \left(\frac{\sqrt{LW}}{r_p} \right)^{D_f}. \tag{2.20}$$

It can be seen that the only changes are that R_G and k_f are exchanged for $\sqrt{LW}$ and k_{LW}, where k_{LW} is connected to k_f.

Continuing with the second estimation which determines the number of particles in a fractal aggregate from the following equation:

$$n_p = k_\alpha \left(\frac{A_a}{A_p} \right)^\alpha . \tag{2.21}$$

A_a is the total projected area of the aggregate and A_p is the projected area of one particle in the aggregate. The number of projected particles are estimated inside the parenthesis and using the constants k_α and α it is expanded to 3D. The values for the constants have been empirically calculated to $k_\alpha = 1.155$ and $\alpha = 1.095$, which according to Wozniak et al. [5] are taken from the most convincing source.

By combining Eq. (2.20) and Eq. (2.21) and taking the logarithm, the following expression is obtained.

$$\log \left[k_\alpha \left(\frac{A_a}{A_p} \right)^\alpha \right] = \log n_p = \log k_{LW} + D_f \log \left(\frac{\sqrt{LW}}{r_p} \right). \tag{2.22}$$

For each image the unknown parameters are A_a, A_p, L, W and r_p, which can be calculated with image analysis. Using multiple images of soot aggregates with the same fractal dimension multiple points will be found in the log-log relation above. The slope of a simple linear regression (see section 2.5.3) of these points will estimate D_f. This restricts the method to only being able to estimate the mean fractal dimension for a set of images of projected aggregates.

The theory above is what was used in the implementation of the MBR algorithm for this master's thesis. Though Wozniak et al. [5] mention something obvious about k_{LW}, it depends on D_f. Connecting Eq. (2.20) and Eq. (2.4) give $k_{LW} = k_f \left(\frac{R_g}{\sqrt{LW}} \right)^{D_f}$. In Eq. (2.22) k_{LW} is considered to be constant which will not be true for all fractal dimensions. In the early stages of this project the MBR method was tested. However since the results were not satisfying and additionally there were time limitations, further tests were stopped at this stage. Now it seems like the problem of a non constant k_{LW}, mentioned in this paragraph, might be the reason. Though further results and discussion concerning this method has been excluded.

2.4.2 Box Counting

The Box Counting method is used to estimate fractal dimension values though it is limited to estimate the fractal dimension up to two dimensions. These estimations will be called the Box Counting dimensions. It is based on Eq. (2.1), which is repeated below.

$$N_L \sim L^{-D_f^{box}}. \tag{2.23}$$

It can be rewritten to the following relation by taking the logarithm on both sides of the proportionality.

$$\log N_L \sim -D_f^{box} \log L. \tag{2.24}$$

The hard thing when implementing this equation in an image is how to calculate the number of sticks N_L with length L needed to measure the object. The method solves this by counting boxes. The image is simply divided into a grid of boxes with a certain side-length L of each box. The number of boxes (N_L) with the object inside are counted for grids with different box-sizes. Now a simple linear regression can be applied to estimate D_f^{box} from the relation in Eq. (2.24). This estimated fractal dimension of the Box Counting methods will be called the Box Counting dimension.

Wozniak et al. [5] mention a problem with the standard-Box Counting method. Dealing with images with a finite resolution result in cropped boxes on the edges of the image. So the total box area counted changes between grids with different box-sizes. They solve the problem by using a Modified Box Counting method which will be called *random-Box Counting* in this report. Here instead of putting the boxes in a grid they are sampled from a uniform distribution to random coordinates in the image. With an equal number of boxes for each scale the area of all boxes will be scale dependent. To solve this the number of counted boxes is normalized by the area of one

box for each scale, transforming Eq. (2.24) to

$$\log \frac{N_L}{L^2} \sim -D_f^{box} \log L. \tag{2.25}$$

Wozniak et al. [5] have used the random-Box Counting dimension as main feature in a machine learning method calibrated with synthesised TEM images of simulated soot aggregates. Their result included a calibration for estimating the fractal dimension from the random-Box Counting dimension and the number of particles in an aggregate. It is shown in Figure 2.2. For each image they used a set of 17 different box side-lengths for the counting. They were [2, 3, 4, 5, 6, 8, 10, 13, 16, 20, 25, 32, 40, 50, 63, 79, 100] pixels wide. To create the calibration they used in their own words *TEM images from thousands of synthetic aggregates*. The aggregates had different fractal dimensions (D_f = 1.2, 1.3, ..., 2.8) and number of particles (n = 5, 10, 30, 50, 100, 400).

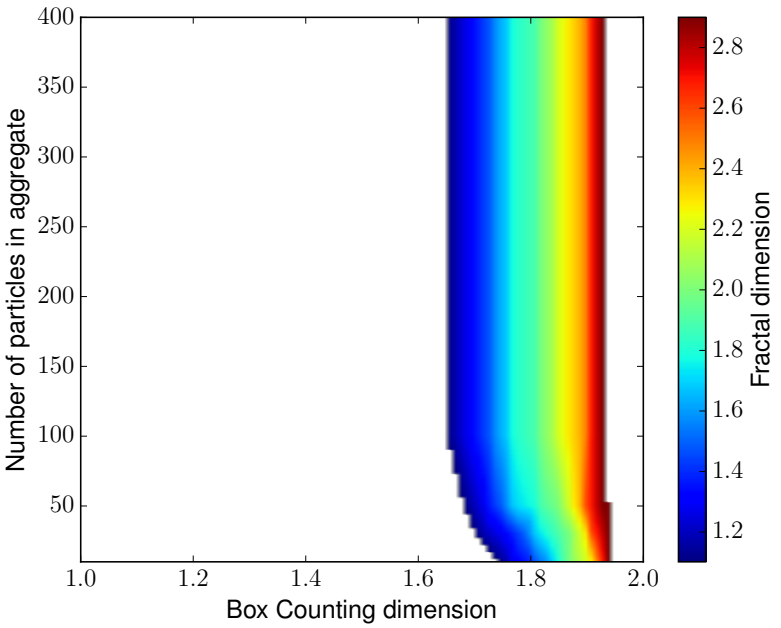

Figure 2.2: The image shows a calibration of the random-Box Counting method produced by Wozniak et al. [5]. With the knowledge of Box Counting dimension and number of particles for an image of a soot aggregate the colour in that point represents the estimated fractal dimension. The image has been produced from images with fractal dimensions [1.1, 1.2, ..., 2.9] and number of particles [10, 30, 50, 100, 400].

2.5 Image analysis methods

As seen in Figure 1.3 a real TEM image of soot has a noisy background level which must be removed prior to the analysis step. This is one of the problems which should be taken care of using the image analysis methods explained below. Implementations of these methods can for example be found in the OpenCV library or the image processing toolbox in Matlab.

2.5.1 Image convolution

In this work 2D discrete convolution, also known as filtering, has been used. If one looks at the 1D case the formula for discrete convolution is the following [12]:

$$(f * g)[n] = \sum_{m=-M}^{M} f[m]g[n-m], \tag{2.26}$$

where the functions f and g only have a finite support in $[-M, M]$. This can easily can be expanded to a 2D version on the form:

$$(f * g)[n, o] = \sum_{p=-M}^{M} \sum_{m=-M}^{M} f[m, p]g[n - m, o - p]. \tag{2.27}$$

Image convolution can be very useful when a specific operation should be performed for each pixel and the neighbouring ones in an image.

2.5.2 Image gradient with pre smoothing

A greyscale image can be seen as a surface in 3D where the third dimension is the greyscale value. When imagined like this it is easy to see that this surface has a slope in both the x and y direction, in other words a vector for the derivative of the greyscale values in x and y direction. Calculating this derivative vector for all pixels in an image the result is called the image gradient. Taking the euclidean-norm of each pixel vector will result in the magnitude of the gradient. The large pixel values in the magnitude of the image gradient correspond to large greyscale changes in the original image. Though taking the derivative is an operation amplifying noise. Therefore it might be preferred to perform noise removal in the image first, by smoothing. A non normalised Gaussian smoothing kernel to convolute with an image is on the form:

$$p(x, y) = e^{-(x^2+y^2)/(2\sigma^2)}, \tag{2.28}$$

where $p(x, y)$ is the value at pixel point x, y in the kernel and σ is a value for the width of the smoothing top. Figure 2.3 shows an example of a Gaussian smoothing kernel.

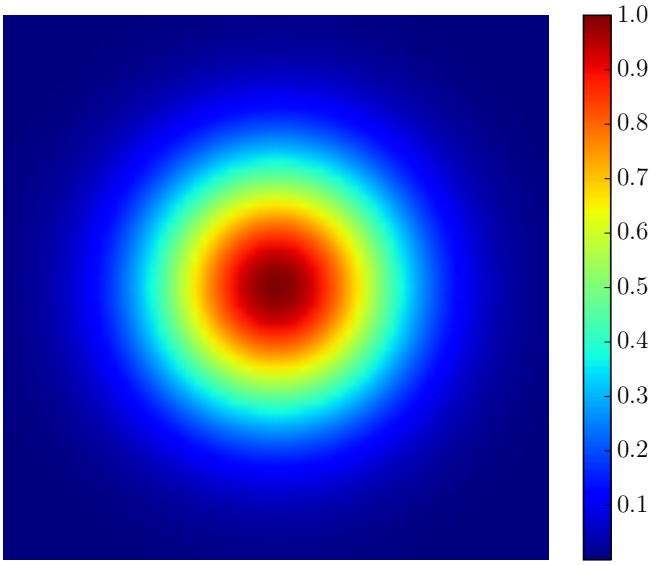

Figure 2.3: Example of a Gaussian smoothing kernel. When convoluted with an image noise is removed and the resulting image is blurried. This example kernel is a 61x61 pixel image with $\sigma = 3$ in Eq. (2.28).

Instead of doing one smoothing and then taking the derivative both can be performed in one kernel. By taking the respective derivative of the Gaussian kernel in p for x and y in Eq. (2.28) the result is:

$$\begin{aligned} p_x &\sim x e^{(x^2+y^2)/(2\sigma^2)}, \\ p_y &\sim y e^{(x^2+y^2)/(2\sigma^2)}. \end{aligned} \tag{2.29}$$

Figure 2.4 shows the derivatives of the kernel in Figure 2.3. Using these kernels in the image convolution will result in the magnitude of the image gradient.

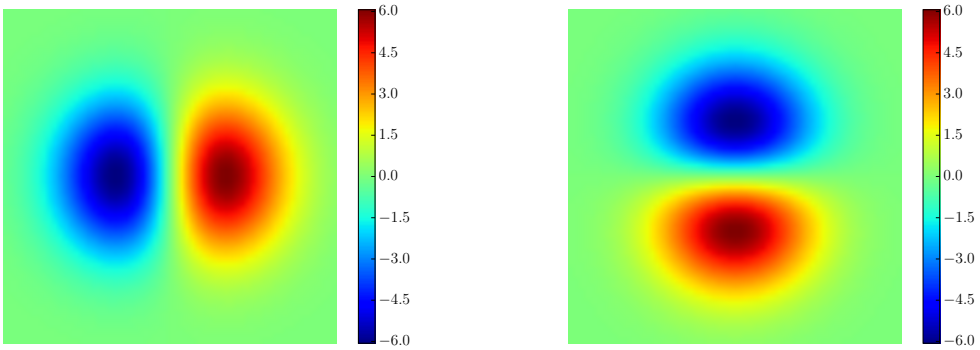

Figure 2.4: Example of a Gaussian smoothing and derivative kernel combined in x- (left) and y-direction (right). These are derivatives of the kernel found in Figure 2.3.

2.5.3 Multiple Linear Regression

From a large number of data points ($\mathbf{y}$, $\mathbf{X}$) a linear model can be estimated to the data. Here $\mathbf{y}$ is some kind of output in the form of a column vector with assumed linear dependence on input values $\mathbf{X}$. $\mathbf{X}$ is a matrix where each row corresponds to input values for output in the same row of $\mathbf{y}$. The model will then be:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.30}$$

where $\boldsymbol{\beta}$ are weights in a column vector with one weight for each input parameter and $\boldsymbol{\epsilon}$ is assumed to be normally distributed, zero mean, noise. The weights $\boldsymbol{\beta}$ should be fitted using multiple linear regression.

There are different ways to find the optimal estimation for $\boldsymbol{\beta}$. One is using the least squares approach where the goal is to choose an estimation $\hat{\boldsymbol{\beta}}$ minimising the loss function:

$$L(\hat{\boldsymbol{\beta}}) = |\boldsymbol{\varepsilon}|^2 = |\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}|^2, \tag{2.31}$$

meaning

$$\hat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} L. \tag{2.32}$$

Taking the matrix derivative [13], setting it to zero, then solving for $\hat{\boldsymbol{\beta}}$ results in:

$$\begin{aligned} \frac{d}{d\boldsymbol{\beta}} L(\hat{\boldsymbol{\beta}}) = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}) = 0, \\ \mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = 0, \\ \hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \end{aligned} \tag{2.33}$$

A simple linear regression is a special case of the multiple one with the model:

$$\mathbf{y} = \beta_0 + \beta_1\mathbf{x} + \boldsymbol{\epsilon}. \tag{2.34}$$

2.5.4 Image threshold

To go from a greyscale image to a binary image with only 0 (black) and 1 (white) as pixel values, the image can be thresholded. The greyscale is often a value in each pixel between 0 and 255. Taking the threshold is simply changing all pixels with a greyscale value above a certain value (called the threshold) to 1. The remaining pixels values are set to 0.

2.5.5 Morphological operations

Morphological operations are important in image analysis and the following is a brief explanation of how they work, the interested reader is encouraged to read further in Jähne [12]. In the work of this master's thesis the operations have only been performed on binary images, meaning that the values of the pixels are either 0 or 1. Here the objects in the image are defined as all pixels with value 1. The morphological operations operate only on one pixel and its neighbours at a time which can either make objects in an image smaller (erosion) or larger (dilation). Erosion and dilation are the two main morphological operations. Both operations need a kernel which in turn is a small binary image describing how much the objects should change. For each pixel in the image the following procedure is done. The middle pixel of the kernel is set on top of the image pixel. All pixels in the kernel are compared to their corresponding ones in the image. If the image is eroded the new pixel value is chosen as 1 only if all 1 values in the kernel are matched with ones in the image. For dilation the new value is chosen to 1 if at least one of the ones in the kernel corresponds to a 1 in the image. For all other combinations the new value is zero. The dilation and erosion procedures are illustrated in Figures 2.5 and 2.6.

The dilation and erosion can be combined. Imagine an erosion followed by a dilation. This is called an opening and for a binary image it will remove noise as seen in Figure 2.7

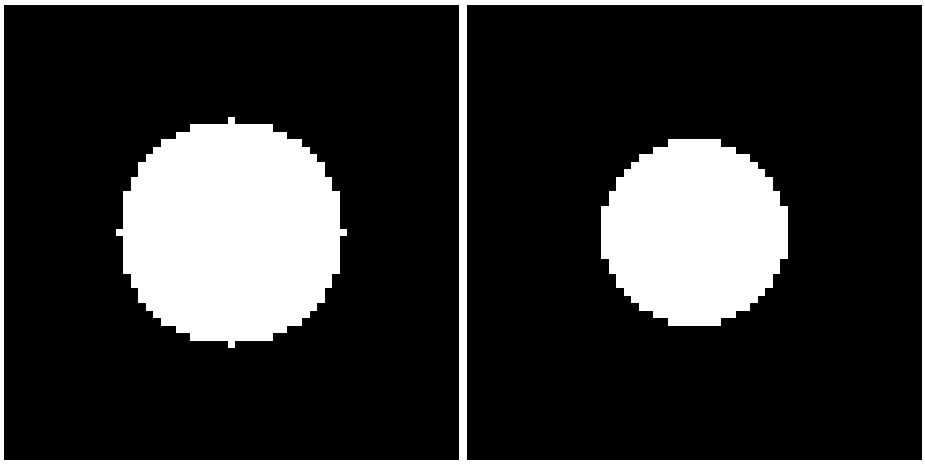

Figure 2.5: The left image is eroded using a 5x5 square pixel kernel resulting in the right one.

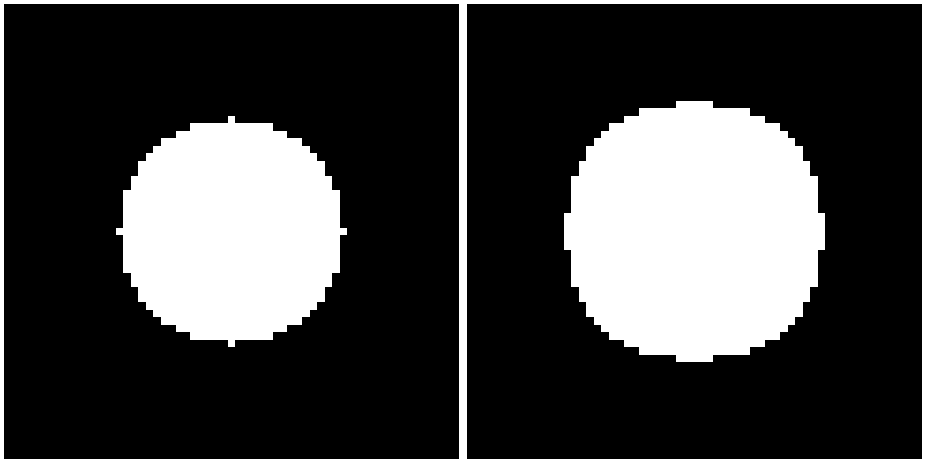

Figure 2.6: The left image is dilated using a 5x5 square pixel kernel resulting in the right one.

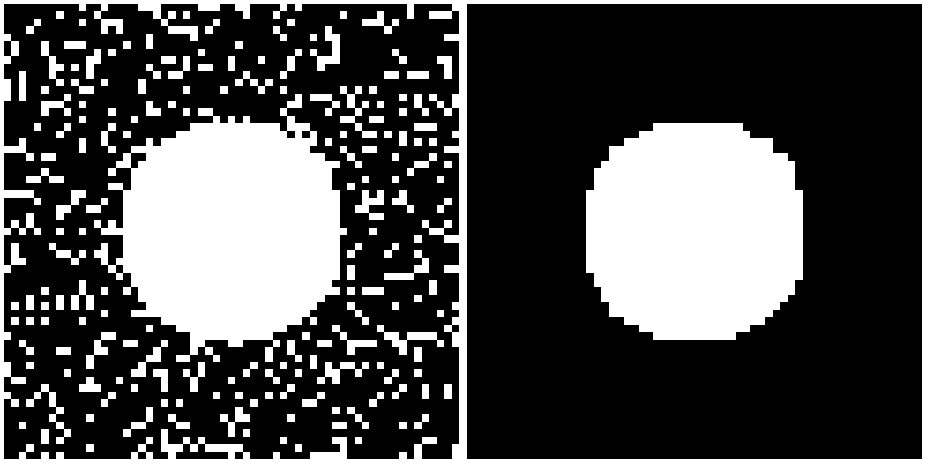

Figure 2.7: The left image is opened using a 5x5 square pixel kernel resulting in the right one.

2.5.6 Finding connected components

The connected components in an image are connected pixels with a similar property. For example in a binary image this property could be white pixels with value 1. Two pixels are connected if they are adjacent in the image. Adjacent could either be four or eight connectivity meaning that either the four closest or the eight closest pixels are classified as adjacent. The connected components for the property of white pixels in a binary image will be *islands* of white surrounded by black.

2.5.7 Filling holes in image objects

In a binary image holes in objects can be filled with image analysis. In this image the background and holes are black and the objects are white. It can be performed in three steps [14].

1. Flood fill the input image from its edges creating a new image. Here the black connected components which also are connected to the image edges are found and their values are changed to white. This image is white everywhere except for the holes.

2. Invert this image, now the holes are the only white parts.

3. Take an *or* operation for each two corresponding pixels in the original and resulting image of the previous step. The result of the *or* operation on two binary pixels is white if at least one of the two pixels are white and black otherwise. Now the background will be black for both images, resulting in black. The object will be white in the original image and black in the inverted, resulting in white. The holes will be black in the original but white in the inverted, also resulting in white.

2.5.8 Finding the minimum bounding rectangle

This problem is also known as finding the minimum area rectangle and is described by Eberly [15]. It finds the general, possibly rotated, minimal rectangle containing a set of points, for example in an image. To do this one can first find the convex hull of all points. This is an ordered subset of the points. The ordered points create a polygon area which includes all points in the set. It can be proved that one of the minimum bounding rectangle's sides must contain one of the edges in the convex hull polygon. One can iterate through all edges and for each, calculate the minimum rectangle containing that edge. Of all these rectangles the one with the smallest area is the minimum bounding rectangle for all points in the set.

Chapter 3

Results and Discussion

Together with the results of extensive simulations, projections and analyses of data, I have during this thesis also taken images of real soot aggregates using a Transmission Electron Microscope. With the help of Thi Kim Cuong Le, Johan Simonsson and Per-Erik Bengtsson, samples of soot were taken from a mini-CAST soot generator on so-called TEM grids. Each grid contained soot from a specific operation point of the generator. The operation points are defined through the relation between three different gas flows to the generator flame; propane, air and nitrogen. They are denoted OP1 to OP7 where OP1 contains the largest most mature soot.

The grids were inserted one by one into the TEM and images of soot aggregates were taken. Among them were 55 TEM images of soot aggregates from Operation Point 1 (OP1), one example is shown in Figure 1.3. These are the images used in the Cluster Analysis of real TEM images.

3.1 Cluster Simulation and Projection

The simulation algorithms described in the theory were implemented and illustrations of four output aggregates from the Cluster-Cluster (CC) method can be seen in Figure 3.1. Corresponding illustrations for the Particle-Cluster (PC) method are found in Figure 3.2. It should be noted how the PC simulated clusters have a more realistic appearance compared to the CC simulated ones. When comparing the clusters with fractal dimension 1.1 for each method the unit lattice grid is intrusive for the CC simulated cluster whereas using continuous coordinates is a more natural approach, seen in the PC simulated cluster.

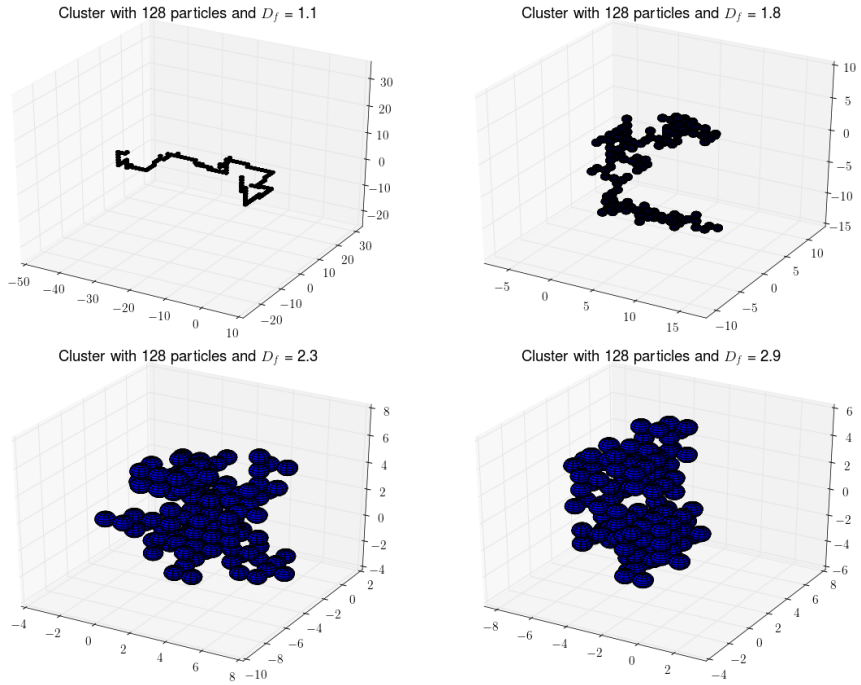

Figure 3.1: Four simulated aggregates using the CC simulation method with 128 particles and different preset fractal dimension. Both are shown in the title of each plot.

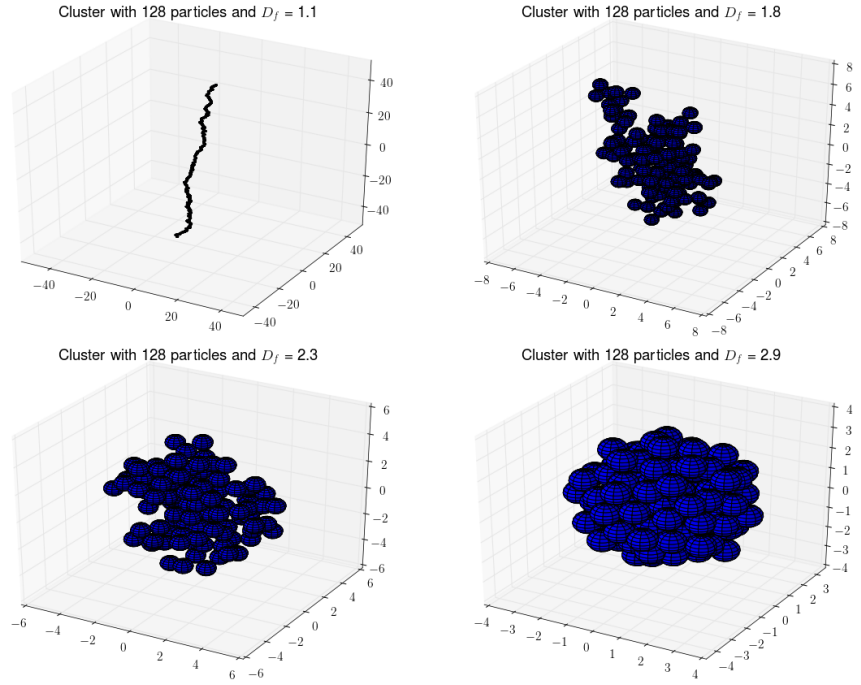

Figure 3.2: Similar to the images in Figure 3.1 but with PC simulated aggregates.

The simulated aggregates are then projected as synthesised TEM images. Examples of the projections of the aggregates in Figure 3.2 are shown in Figure 3.3. These are binary black and white images which are not realistic TEM images compared to the TEM image in Figure 1.3. The choice of binary projection output is based on the fact that the Box Counting methods, described in section 2.4.2, take binary images as input. With projection directly to binary images the preprocessing step for each image prior to the analyse methods can be removed. The drawback

of unrealistic projections is therefore more than compensated for by faster execution time.

The projection implementation first takes a random projection angle as a 3D vector. An affine camera matrix is created with the direction of this vector looking at the centre of gravity of the cluster. The affine camera is motivated as a valid approximation from the large amplification used in a TEM. With the camera matrix the points of all particles in the simulated aggregate are projected into a 2D plane. The minimum bounding rectangle of these points are calculated and all points are rotated to have the sides of this rectangle parallel to the x- and y-axis. The rotated points are transformed using an intrinsic matrix with the scale and image centre calculated for the whole cluster to fit in a 1028x1028 pixel image. With the coordinates of each particle in the image, they are inserted one by one as filled circles of white with a radius matching the 3D overlap used by Wozniak et al. [5], mentioned in section 2.3. The aggregate in the projected image has been rotated to its minimum bounding rectangle, though with the constant value of 1028x1028 pixels there might be redundant black pixels close to the edges. These pixels are removed as the final step in the projection algorithm.

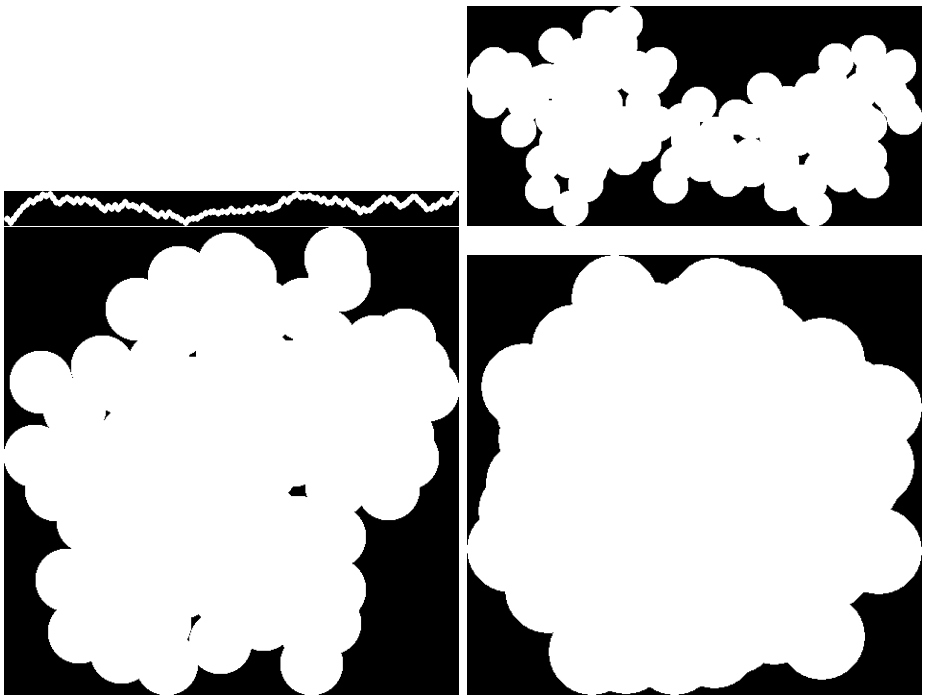

Figure 3.3: One projection of each aggregate shown in Figure 3.2.

The projected image of an aggregate is largely dependent on the projection angle. From just one or two images it is hard to make any conclusions about the general projection behaviour of aggregates with a certain number of particles and fractal dimension. To get around this problem, large datasets of projected aggregates were produced. Each dataset consists of projections of aggregates with equal number of particles and fractal dimension. Combining the output of an analysing method for a whole dataset will give a more general description of the projection behaviour in it.

Each dataset consists of 1000 projected images of simulated aggregates. All aggregates are projected once. For the CC method, datasets with $[16, 32, 64, 128, 256]$ particles were produced, due to the restriction in number of particles. The PC method instead used the particle numbers $[10, 30, 50, 100, 400]$, mimicking the ones used by Wozniak et al. [5] in their calibration, which is shown in Figure 2.2. For each number of particles and simulation method, sets with preset fractal dimensions $[1.10, 1.11, 1.12 \ldots, 2.89, 2.90]$ were created.

The choices for all numbers in the last paragraph is a balance between execution time, maximising dataset size and precision of results. The precision of results, is connected to the choice of the fractal dimension interval, the lowest and largest values are excluded. When they were included it resulted in impossible calibrations. For these fractal dimensions the Box Counting dimensions were no longer strictly increasing with the fractal dimension. Meaning the same Box Counting

dimension corresponded to multiple fractal dimensions. Since none of the real soot aggregates seem to have fractal dimensions outside the range $[1.1, 2.9]$ these datasets were excluded.

Together, all datasets consist of 1 810 000[1] synthesised TEM images from an equal number of simulated aggregates. The total execution time simulating and projecting these datasets is less than 12 hours.

3.2 Cluster Analysis with Box Counting

As explained in the theory section 2.4.2, the Box Counting methods counts boxes. This is done for boxes with different sizes where only the boxes with the object inside are counted. An example of how the boxes are counted is shown in Figure 3.4 for the top right aggregate in Figure 3.3. At the right and lower edges of the images the cropping of boxes in the grid is seen, mentioned in the theory of the standard-Box Counting method.

With the counted boxes a line is fitted to the log-log plot of the number of boxes to the side-lengths of the boxes for a single image. Figure 3.5 shows this plot for the image used in Figure 3.4. The same box side-lengths as Wozniak et al. [5] have been used. There is also a modified version of the method called random-Box Counting. Examples of the random-Box Counting boxes are shown in Figure 3.6 where the corresponding log-log plot is shown in Figure 3.7.

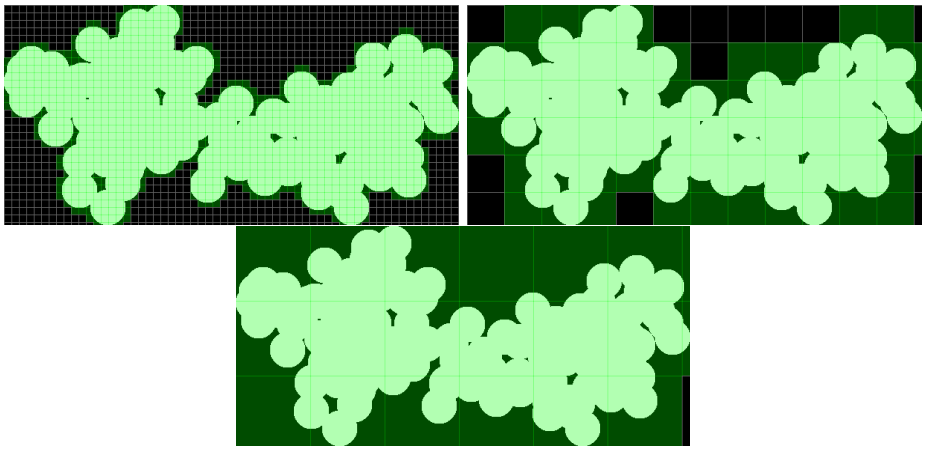

Figure 3.4: A cluster with three different standard-Box Counting grids in foreground. The green boxes are the ones counted in the method.

[1] 2 (simulation methods) x 5 (number of different number of particles) x 181 (number of different fractal dimensions) x 1000 (simulations and projections per dataset)

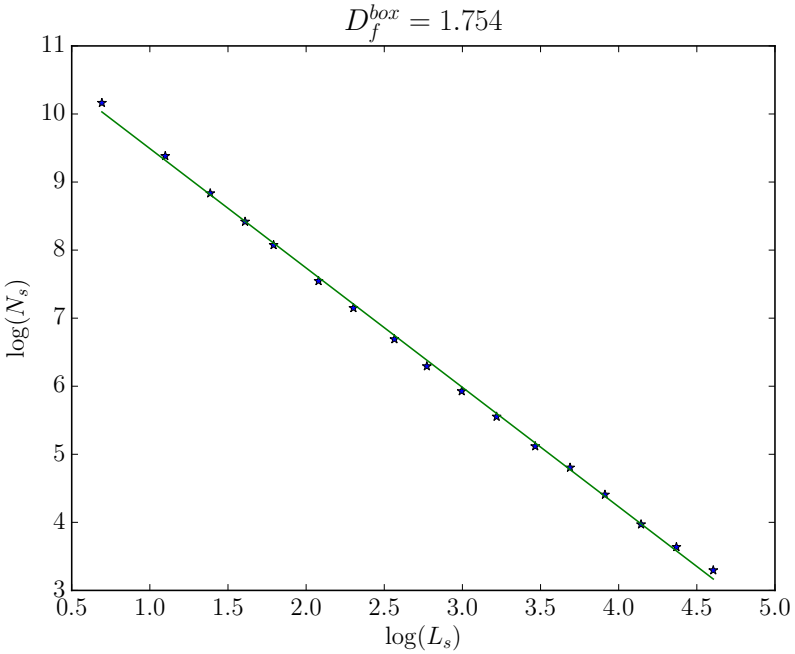

Figure 3.5: A log-log plot of the standard-Box Counting method's number of boxes to the side-length of the boxes. The resulting estimated Box Counting dimension is shown in the title of the plot. It is equal to the slope of the line fitted to the points.

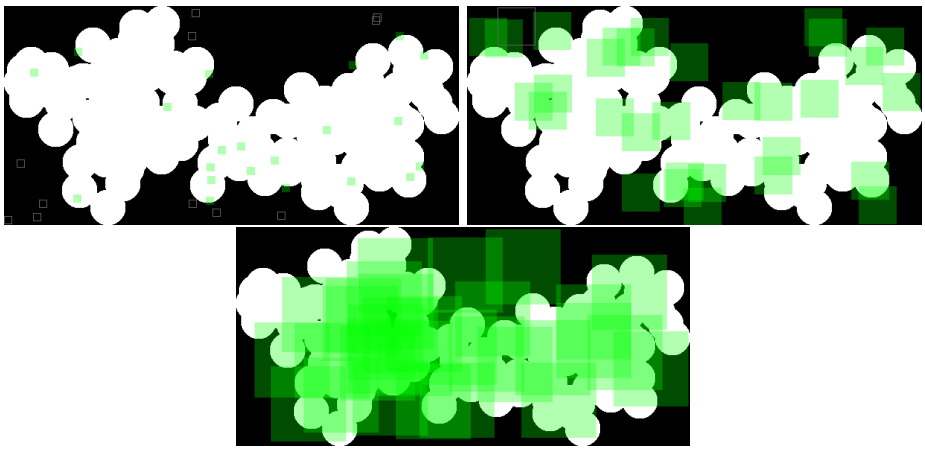

Figure 3.6: Illustration of the random-Box Counting method in a similar way as Figure 3.4. When used for the real calibration and analysis 2^{15} boxes are used compared to the 32 ones seen in these images.

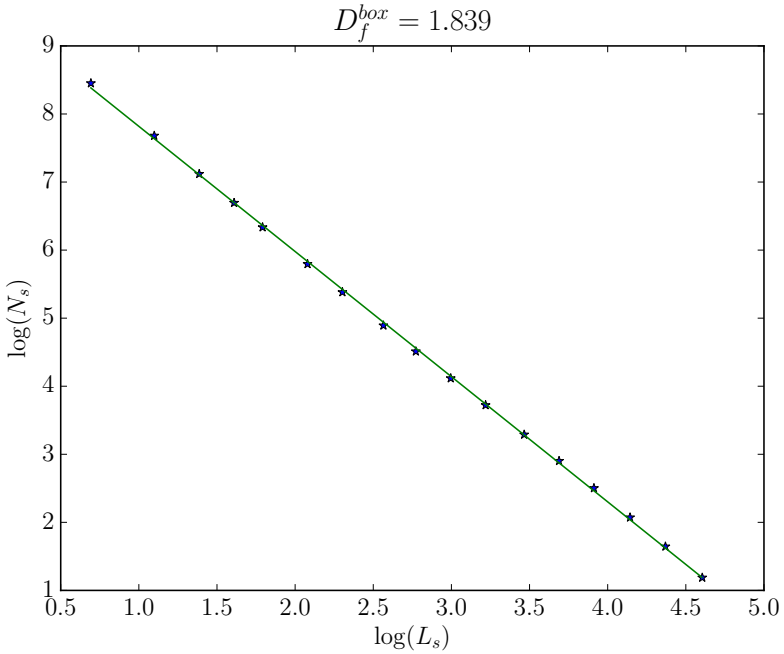

Figure 3.7: The same plot as in Figure 3.5 but for the random-Box Counting method.

The Box Counting methods only produce an estimation for the fractal dimension in 2D as mentioned. To avoid confusion it is important to remember that there are two methods and there are outputs from these methods. The methods will be called standard-Box Counting and random-Box Counting. Though the output 2D estimate of the fractal dimension will for *both* methods be called the Box Counting dimension.

The Box Counting dimension was used as a feature in a machine learning method estimating the 3D fractal dimension. Though alone it will not suffice. The number of particles in an aggregate also has a large influence on the input projections. It is easy to visualise a projection of an aggregate with just a few particles and the difference to one with many. The projection of the one with fewer particles is more probable to be perceived as having a larger fractal dimension, and Box Counting dimension, compared to the corresponding projection with more particles. Therefore the number of particles in an aggregate is also included as a feature.

When the Cluster-Cluster simulation method was used in the analysis the result reflected the unrealistic appearance of its aggregates. This is the reason for why further results and discussion of this method have been excluded.

For a single dataset there will be 1000 different Box Counting dimensions. To understand the Box Counting methods it is important to see how these values are distributed, both for a single dataset and for datasets with different fractal dimensions. These distributions can be described statistically with the stochastic variables D_f for the fractal dimension and D_f^{box} for Box Counting dimension. Plots estimating the 2D intersection probability density function, $P(D_f \cap D_f^{box})$, have been produced which will be called uncertainty plots. They are shown in Figures 3.8 and 3.9 where the distributions have been estimated for PC datasets with 400 particles. The difference between them is that Figure 3.8 is produced with the standard-Box Counting and Figure 3.9 with the random-Box Counting method. The estimations are calculated with a 2D histogram method.

From the uncertainty plots it can be seen that the random-Box Counting has a wider distribution of Box Counting dimensions for each fractal dimension, especially for the lower ones. Intuitively this might seem like a bad property of the random-Box Counting method. Though from the possible projections of an aggregate with low fractal dimension this wider distribution is quite reasonable. If the projections in the dataset with the lowest fractal dimension and 400 particles are studied there are images looking in the direction of the particle chain of the aggregates. The chain can be seen in the top left projection in Figure 3.3. These projections are in the most extreme case just a dot which is perceived to be aggregates with a large fractal dimension,

they should be reflected as large Box Counting dimensions. It seems like these images have been incorrectly classified as too low Box Counting dimensions with the standard-Box Counting method. The random-Box Counting method has more reasonable estimations.

The uncertainty plots creates the possibility to calculate the distribution of fractal dimension values conditioned on the Box Counting dimension:

$$P(D_f|D_f^{box}) = \frac{P(D_f \cap D_f^{box})}{P(D_f)}. \tag{3.1}$$

This is applied by choosing a Box Counting dimension on the x-axis and following the colour distribution along the y-axis. Two plots for the conditioned distributions have been produced for each Box Counting method. The distributions are conditioned on one of their respective lowest and largest Box Counting dimensions. The plots are shown for the standard-Box Counting method in Figure 3.10 and random-Box Counting in Figure 3.11, together with estimations of the standard deviation of these distributions. The standard deviation can be interpreted as the uncertainty of the fractal dimension estimation for the conditioned Box Counting dimension. Larger standard deviation corresponds to a larger uncertainty. Mutual for both methods is a larger uncertainty of the fractal dimension estimation for large Box Counting dimensions compared to the lower ones. This is reasonable. Connected to the previous discussion about projection angle a majority of aggregates can be projected from a direction to look like an aggregate with a large fractal dimension. These projections will produce large Box Counting dimensions. In contrary only aggregates with low fractal dimensions can be projected to look like an aggregate with low fractal dimension which will correspond to low Box Counting dimensions. Inverting this reasoning gives that low Box Counting dimensions can only correspond to low fractal dimensions where large Box Counting dimensions can correspond to a wide interval of fractal dimensions. This is reflected in the conditioned probabilities in Figures 3.10 and 3.11. The large uncertainty for high Box Counting dimensions is a result of losing the third dimension when projecting a 2D image of a 3D aggregate.

The uncertainty for estimating the fractal dimension from the conditioned plots is also compared between the Box Counting methods. The values show that the standard-Box Counting method has a lower uncertainty than the random-Box Counting method, but the differences are slim. It might be that the differences are not large enough to be statistically significant.

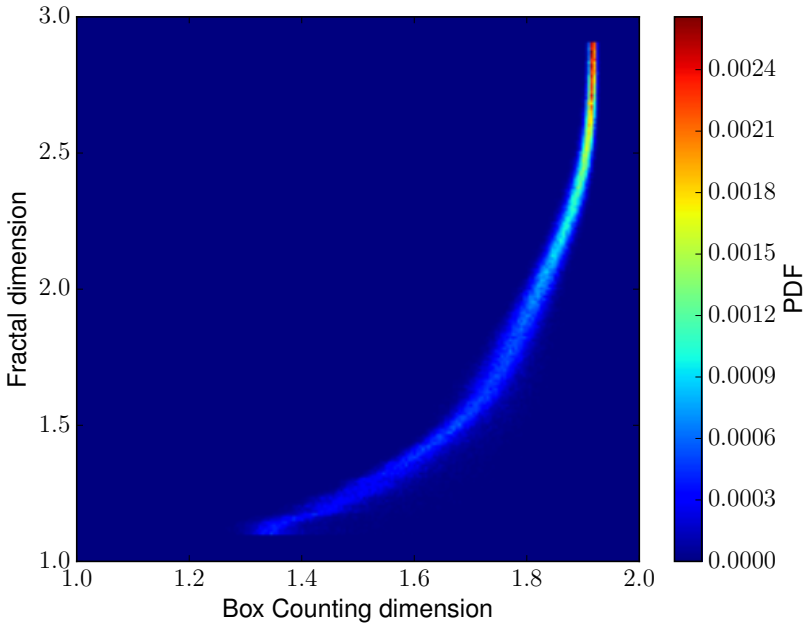

Figure 3.8: The so called uncertainty plot which is an estimation of the intersection probability density function, $P(D_f \cap D_f^{box})$. Here D_f is the stochastic variable for the fractal dimension and D_f^{box} is the stochastic variable for the standard-Box Counting dimension. The x-axis is the Box Counting dimension, the y-axis is the fractal dimension and the colour is the probability. It has been calculated for the datasets with 400 particles.

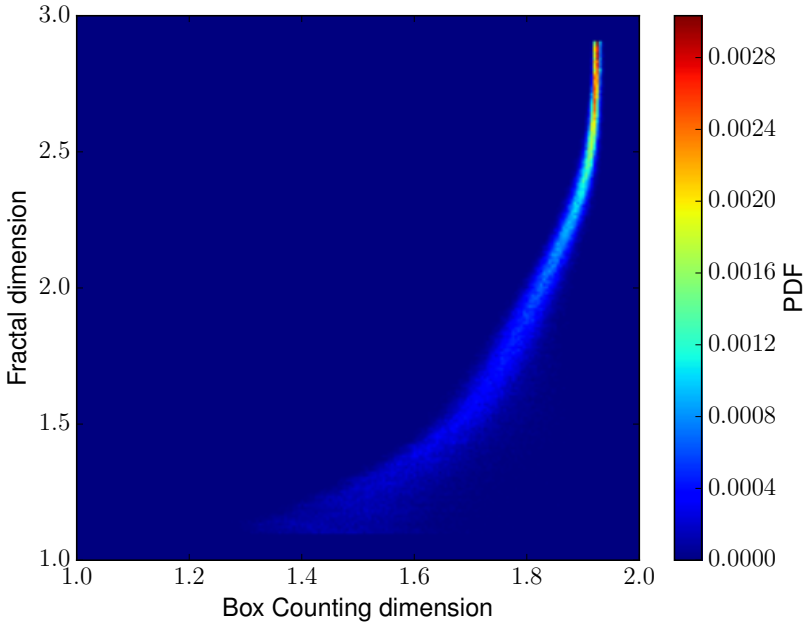

Figure 3.9: The same plot as in Figure 3.8 with the difference of using the random-Box Counting method.

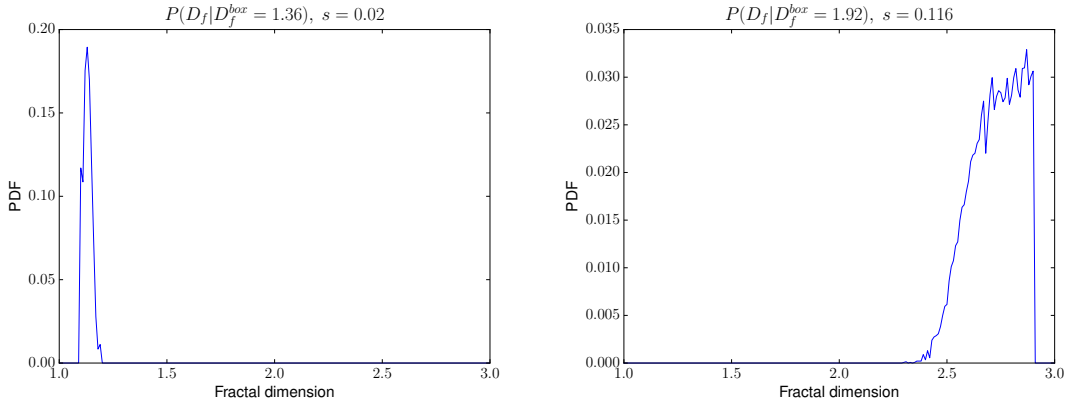

Figure 3.10: These plots are Probability Density Functions for the fractal dimension conditioned on two different Box Counting dimensions. The conditioned Box Counting dimension together with an estimation of the standard deviation s of the distributions are shown in the titles. The plots are calculated from the uncertainty plot in Figure 3.8.

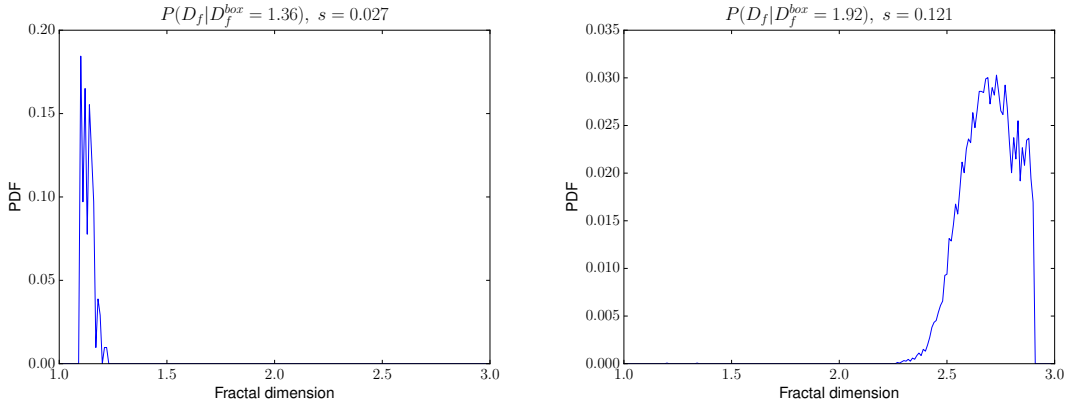

Figure 3.11: The same plots as in Figure 3.10 with the difference of using the uncertainty plot for the random-Box Counting method in Figure 3.9.

Each dataset, with a certain number of particles and fractal dimension, will result in 1000 different Box Counting dimensions. The mean of all these values for each dataset is calculated. This is assumed to be the general Box Counting dimension for the combination of number of particles and fractal dimension which results in a 3D point for each dataset. The coordinates are: Box Counting mean dimension, number of particles in aggregate and fractal dimension. These points for all datasets are used to calibrate the machine learning method where the calibration is represented by a colour plot. The x-axis are Box Counting dimensions, y-axis are the number of particles and the colour is the fractal dimension value. The colours of the area between the 2D points of all datasets are filled using linear interpolation. For a new image with unknown fractal dimension the Box Counting dimension can be calculated and number of particles counted. The colour of this point in the calibration plot is the estimated fractal dimension. There will also be invalid white coloured points where no estimation of fractal dimension is possible.

There are two different analysis methods which will each have a calibration when used with the PC datasets. The standard-Box Counting calibration is found in Figure 3.12 and random-Box Counting calibration in Figure 3.13. General comments on the calibrations are that larger Box Counting dimensions correspond to larger fractal dimensions and aggregates with fewer particles have larger Box Counting dimensions for most of the fractal dimensions. Both of these properties are consistent with previous discussions and this correlation between the Box Counting dimension and the fractal dimension clearly show that the Box Counting dimension extracts the fractal properties of the projection of an aggregate.

When comparing the calibrations of the analysis methods in Figures 3.12 and 3.13. There are two clear differences. The random-Box Counting method seems to translate the Box Counting dimensions to larger values and decrease the Box Counting dimension span. Just from the calibrations it is hard to say anything about which one is better than the other. It will be revisited in the discussion of the Cluster Analysis of real TEM images in section 3.4.

This thesis has reproduced the method of Wozniak et al. [5]. They used the random-Box Counting method together with synthesised TEM images projected from PC simulated clusters. Therefore the PC calibration with random-Box Counting in Figure 3.13 should be compared to the corresponding one in Figure 2.2 by Wozniak et al. [5]. The implementation in this report seems to result in a smaller span of Box Counting dimensions for the lowest number of particles and a wider one for the largest number of particles. Another difference is a much smoother colour gradient for each number of particles in Figure 2.2. This is preferred since it will give a lower uncertainty in fractal dimension estimation depending on the Box Counting dimension. Since both images should have been produced with the same methods it is interesting to discuss why they differ which could be connected to either the synthesising of TEM images or the random-Box Counting method. After looking at thousands of synthesised TEM images produced in this project they seem to be realistic. Both the different number of particles and fractal dimensions looks to be fulfilled in the output images. This leaves the question of whether the random-Box Counting methods are equal. All results from the implementation in this report show nothing suspicious about it and there is no ambiguity concerning their explanation of the method. It is hard to know what creates the difference without comparing their implementation to the one used in this report. It should also be noted that even though the calibrations are different they are also very similar.

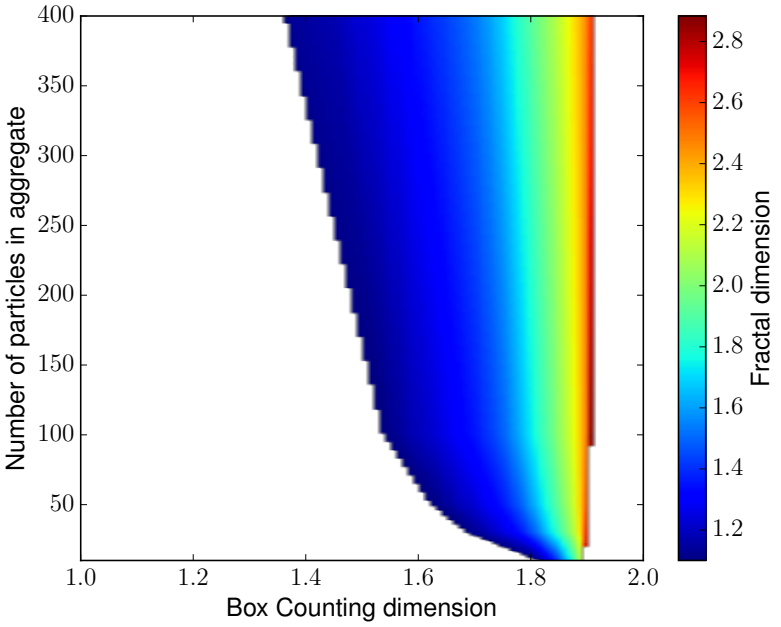

Figure 3.12: The calibration of the standard-Box Counting method. Here the datasets from the Particle-Cluster simulation method have been used. For an image of a soot aggregate with calculated Box Counting dimension and counted number of particles, the fractal dimension can be estimated as the colour in that point in the image.

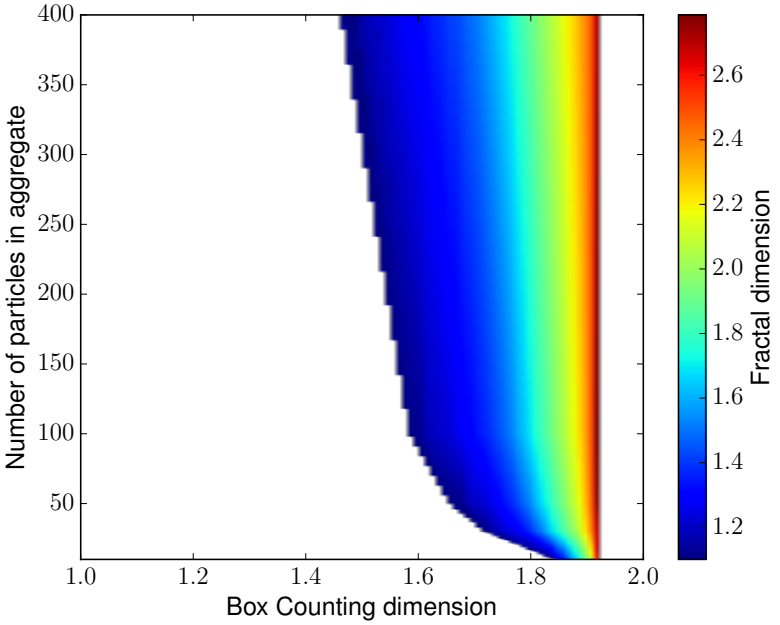

Figure 3.13: A calibration as in Figure 3.12 but with the random-Box Counting method.

3.3 Preprocessing of real TEM images

The final pipeline of the preprocessing will be visualised using the soot cluster found in Figure 1.3. The goal of the preprocessing is to get a binary image where only the white parts are cluster and everything else is black. To make the rest of the visualisation more intuitive the inverse of the image is taken, making the brighter parts cluster just like for the expected output. This inverted image is found in Figure 3.14. For each cluster the number of particles are also counted by hand. Here it should be noted that this is a very subjective approach of attempting to calculate the number of particles in 3D from a 2D projection.

To end up with a preprocessed binary image one method is to threshold it. Looking at the inverted image, in Figure 3.14, this will be difficult. Comparing the pixels in the corners of the image, the top left ones seem to be brighter than the lower right. The continuous change in background light affects both the background and the cluster in the image. Meaning, a good threshold value will be difficult to find for the whole image. In the end the continuous change in background light must be removed. This is done in three steps. First a rough background segmentation is performed, giving the result found in Figure 3.15. The background segmentation is a separate method with the following steps. First the magnitude of the image gradient is calculated and thresholded. This will find almost all edges of the clusters. The result is dilated with a quite large kernel to fill up holes in these edges. The rest of the holes in the clusters are then filled with the method described in section 2.5.7 resulting in a rough cluster estimate where the inverse is a rough background estimate. The second part of removing the continuous change in background light is to use the pixels classified as background in a multiple linear regression model. A few regression models were tested and the best one was a second degree, 2D polynomial on the form:

$$g(r, c) = a_0 + a_1 r + a_2 c + a_3 r^2 + a_4 c^2 + a_5 rc, \tag{3.2}$$

where $g(r, c)$ is the background greyscale value for pixel-row r and pixel-column c and a_x are model parameters to be estimated. The third step use the parameters calculated from the regression and computes a new image with only the estimation of the continuous change in background light. For the example it will, just like in the inverted image, be brighter in the top left corner and darker in the lower right one. This image is subtracted from the real image. The result is shown in Figure

3.16, and reusing the rough background segmentation leaves the problem of finding the foreground in Figure 3.17.

Back to the first problem of finding a good threshold for the image. Now what is a good threshold? When a threshold was optimised for one image it apparently only sufficed for that one, meaning a more generic solution was needed. Since the problem is to find a greyscale value the distribution of the greyscale pixel values was estimated. The distribution can be estimated as the histogram of the pixels greyscale values which is found in the left plot of Figure 3.18, the histogram value of the black pixels has been excluded since it is huge. Looking at the histogram these two Gaussian-like peaks were found for almost every image which seem to describe the image quite well. The first peak from the left mostly corresponds to the noisy background pixels and the second one the cluster pixels. The middle of the first Gaussian-like peak is chosen as a compromise between getting as much cluster as possible while still restricting noise. The peak is found as the global maximum of the histogram after the following processing. First it is smoothed using a median smoothing filter, removing narrow peaks. Then the second peak often was larger than the first one. This was fixed by lowering it from the multiplication with the following linear function,

$$f(x) = 1 - \frac{x}{255}, \tag{3.3}$$

where x is the greyscale value on the x-axis in the histogram plot. Now the right plot in Figure 3.18 has been calculated and the local maximum of the first Gaussian-like peak is also the global maximum of the whole histogram.

The threshold value calculated in the previous part is used with the image in Figure 3.17 and the thresholded image is shown in Figure 3.19. The resulting image has a noisy background close to the aggregate. This noise is removed using the morphological opening operation which also removes parts of the aggregates. To compensate for this and retrieve optimal results a small dilation is also applied to the image. This results in the image in Figure 3.20. Using the method of finding the connected components with four connectivity, described in section 2.5.6, will now solve two problems. Too small clusters which mostly are remnants of noise can be removed. Also multiple clusters in the same image are separated and the final segmentation is shown in Figure 3.21. The two clusters are saved, rotated and cropped to their minimum bounding rectangle, in different files.

The preprocessing was applied to the 55 TEM images taken on OP1 soot clusters. It managed to find 52 different soot clusters. These processed clusters are, together with the counted number of particles for each one, the results of the preprocessing of the real TEM images.

The preprocessing algorithm could not produce a viable segmentation for some clusters. This mostly depended on the histogram not following the assumed pattern of Gaussian-like peaks and the wrong threshold was found. Two more examples of preprocessing output are shown in Figure 3.22. The left one is an example of a good cluster segmentation though the right one is poor. At the edges of this aggregate, small circles of white can be found which seem to be out of place. These are the result of noise remaining after the opening operation. The noise have been dilated and connected to the main aggregate component in the image and is thereby classified as aggregate in the final segmentation. The right image in Figure 3.22 is a cluster with fewer particles, for which the preprocessing in general outputs poor segmentations.

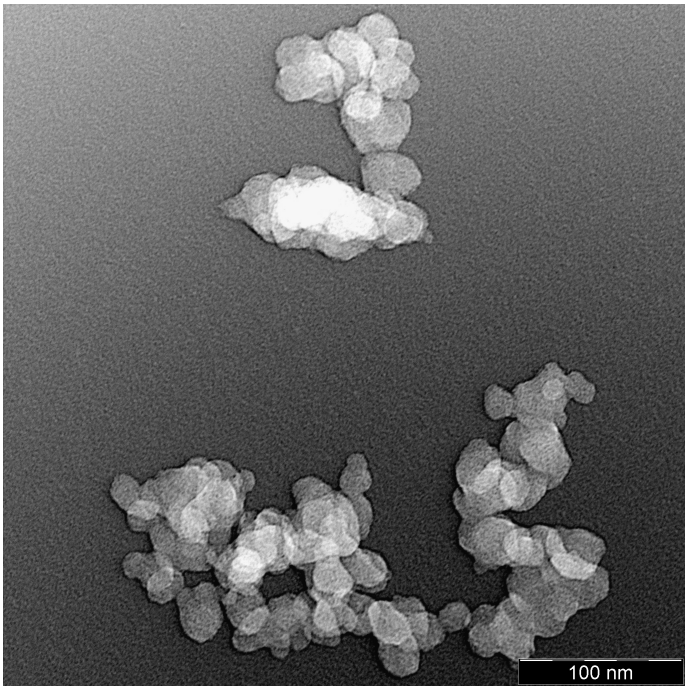

Figure 3.14: The inversion of the image shown in Figure 1.3

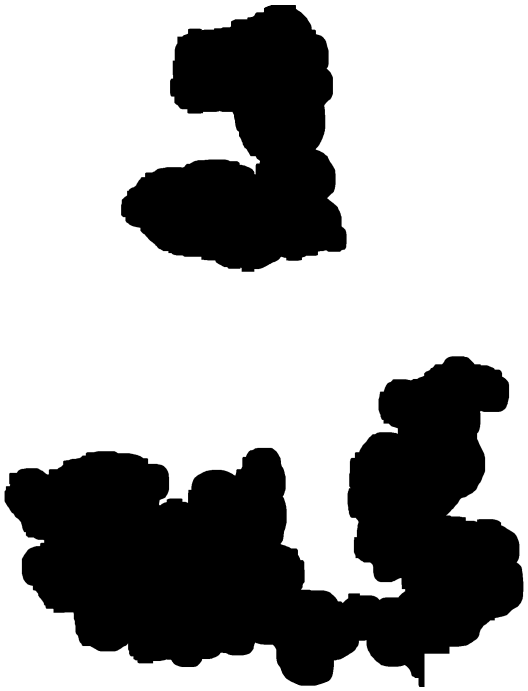

Figure 3.15: The white pixels in this image are together the rough estimation of the background pixels in Figure 3.14. They were found using a thresholded gradient followed by a dilation and finalised with a filling of possible white holes in the black parts of the image.

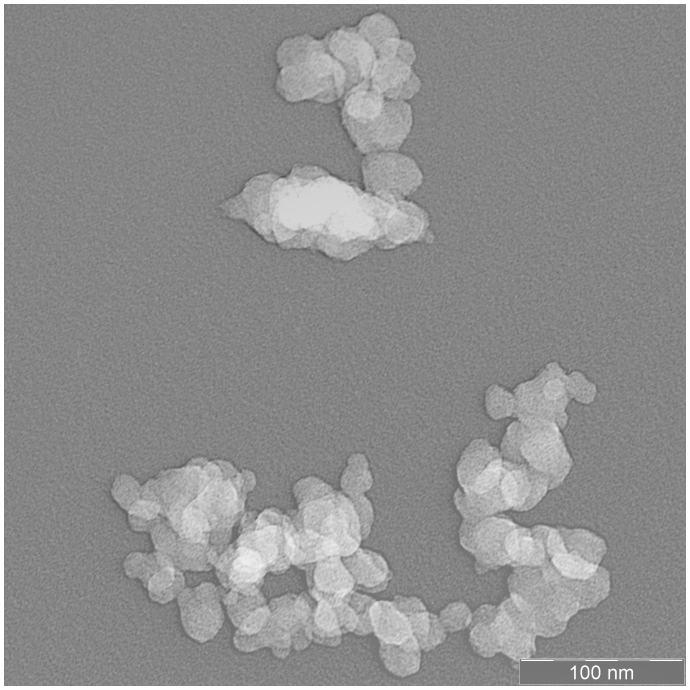

Figure 3.16: The result after the continuous change in background light has been subtracted from the image in Figure 3.14.

Figure 3.17: The remaining pixels to classify as non-black after the normalisation. A combination/multiplication of the inverse of the image in Figures 3.15 and the image in Figure 3.16.

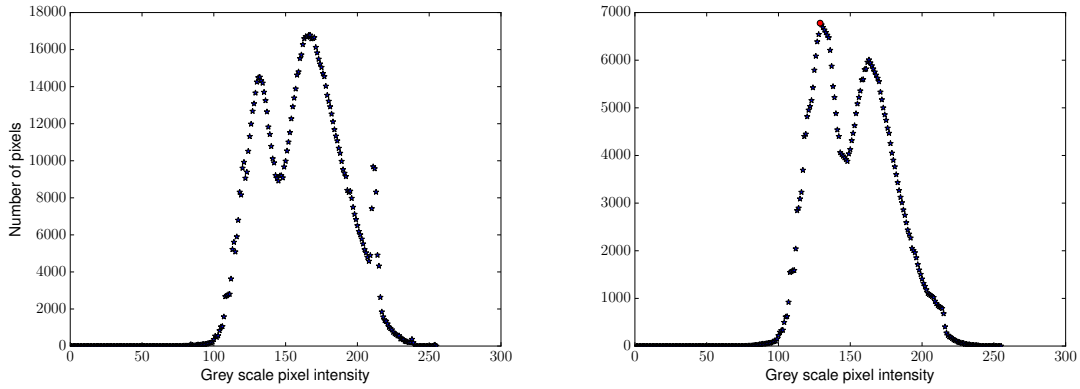

Figure 3.18: The left image shows a histogram of the pixels greyscale values from the image in Figure 3.16. The x-axis in the plots represents the greyscale pixel intensities. The number of pixels with zero value (black) are excluded since they are both to many and not interesting. The right plot shows the output after processing the histogram in the left plot. This finds the Gaussian-like peak with the lowest greyscale value as the global maximum (red dot).

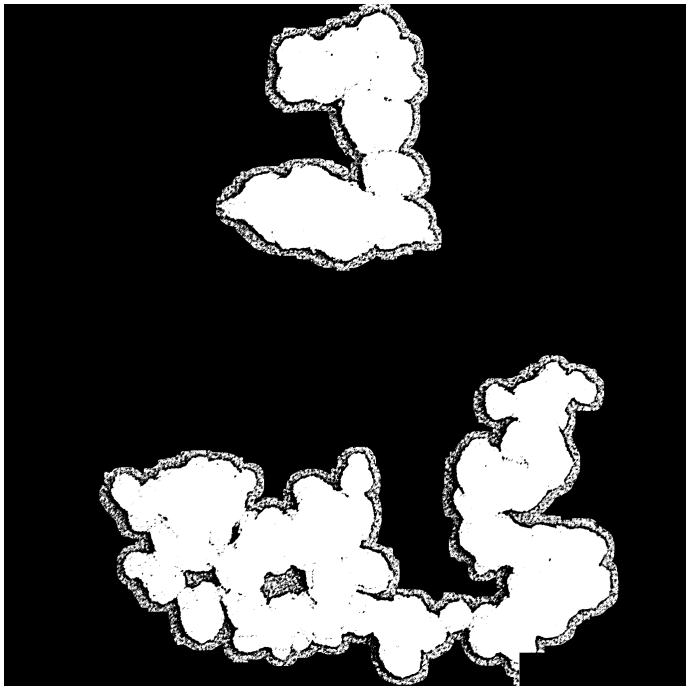

Figure 3.19: Resulting image after using the found threshold in Figure 3.18 on the image in Figure 3.17.

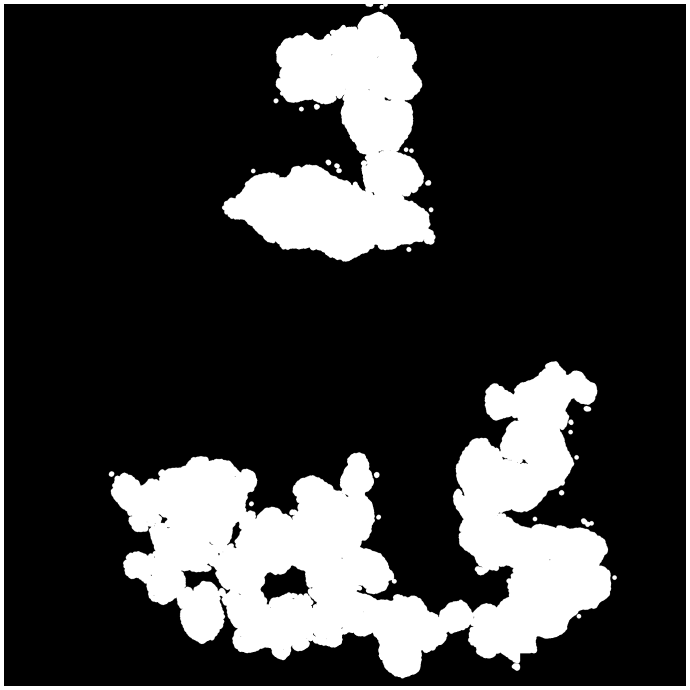

Figure 3.20: The image in Figure 3.19 after noise removal by the morphological opening operation followed by a small dilation.

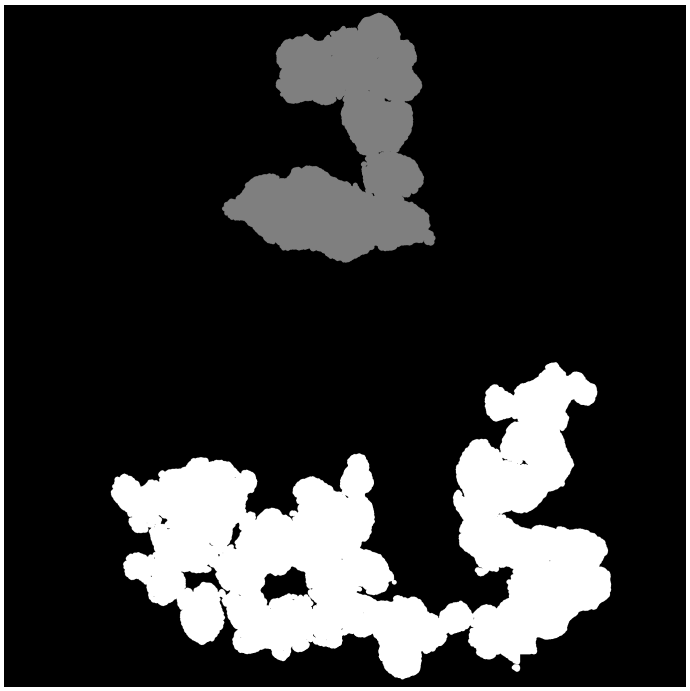

Figure 3.21: The resulting clusters found as two different ones and saved to different files rotated and cropped to their respective minimum bounding rectangle.

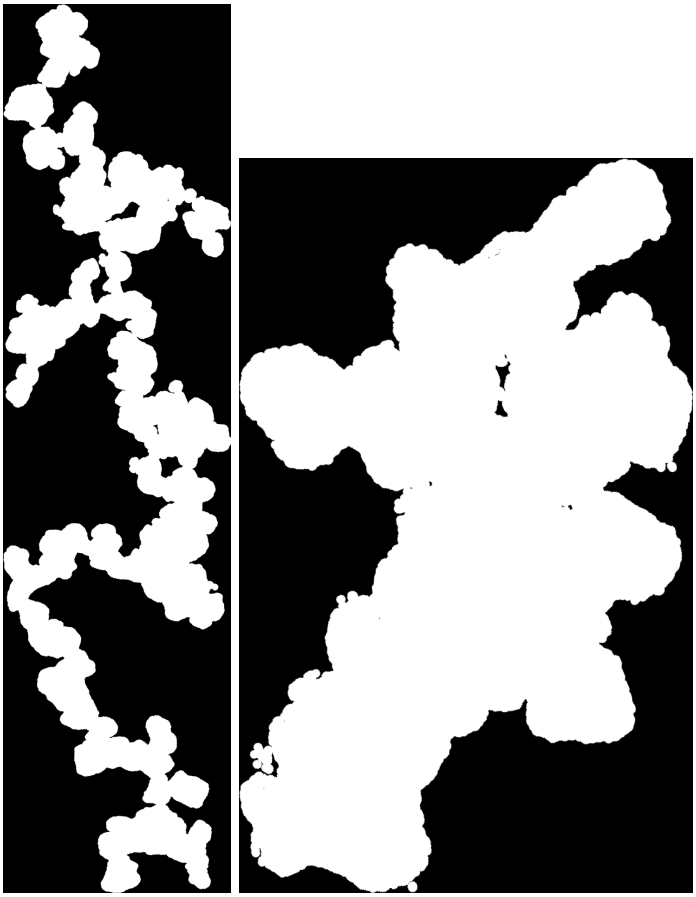

Figure 3.22: The result of preprocessing two real TEM images of OP1 soot clusters.

3.4 Cluster Analysis of real TEM images

From the preprocessing, the analysis of 52 soot clusters now awaits. For the standard-Box Counting and random-Box Counting method, example results for four different aggregates are shown in Figures 3.23 and 3.24. Here the Box Counting dimension, number of particles and estimated fractal dimension are shown in the image titles. To get a feeling of the result for all the real soot aggregates two visualisations have been produced. First plots visualising the dependence between the counted number of particles and the estimated fractal dimension of real TEM images are shown in Figure 3.25 for standard-Box Counting and Figure 3.26 for the random-Box Counting method. Their points have also been plotted in the calibrations from PC simulation and Box Counting methods in Figures 3.27 and 3.28. At last the number of real soot images valid to each calibration and the mean of all estimated fractal dimensions for the different methods are summarised in Table 3.1.

The values in Table 3.1 can be compared to the ones from previous works in Table 2.1. The results in this report are generally larger than the ones in previous works, especially when comparing the corresponding values of 1.94 and 1.66. One of the more obvious possible reasons for the larger mean of estimated fractal dimensions is the preprocessing step. It might alter the real TEM images in a way, which increase the Box Counting dimensions compared to the preprocessing methods of Wozniak et al. [5] for example. The preprocessing also included manual counting of particles which might have a large uncertainty. Another reason might be connected to the soot aggregates used in the Cluster Analysis of real TEM images. They have been sampled from the cold gas exhaust of a mini-CAST soot generator at Operation Point 1. This should be compared to the sampling inside flames by Wozniak et al. [5] and Dobbins and Megaridis [6]. There is a possibility that the soot aggregates fold themselves when cooled in the exhaust of the generator, thereby obtaining a larger fractal dimension. This means that the soot aggregates in the TEM images studied in this

master's thesis could actually have a larger fractal dimension compared to the ones in previous works.

The plots in Figures 3.25 and 3.26 visualise the dependence between the fractal dimension estimations and number of particles for the Box Counting methods. For both methods there seem to be a larger variation in estimations for fewer particles which decrease with the increase of particles. The shape, and fractal properties, of an aggregate with few particles depends more on the location of every single particle compared to one with more particles. Since there always will be a variation of the particle locations for aggregates the shape of aggregates with fewer particles will vary more. Therefore the result of a larger variation of estimated fractal dimensions for aggregates with fewer particles seems valid. Otherwise no specific differences between the Box Counting methods are found, excepting generally lower estimations of fractal dimensions for the random-Box Counting method.

The standard-Box Counting method have a few invalid points. They are found as the points in white parts of the calibration in Figure 3.27, where they all seem to have too large Box Counting dimensions. This could be connected to soot aggregates without fractal properties. Wozniak et al. [5] have assumed that aggregates with fewer than 10 particles lack fractal properties. Though looking at the points in Figure 3.27, one invalid aggregate has at least 50 particles. This aggregate should have fractal properties, meaning that there is a suspicion of the Box Counting dimension for the standard-Box counting method on these TEM images to be wrong.

Table 3.1: Summarising the number of valid soot aggregates (N) of the original 52 and the mean of all the estimated fractal dimensions (D_f) from the use of different estimation methods. Invalid points are all combinations of N and Box Counting dimensions corresponding to white points in the calibrations.

Calibration and Cluster Analysis method	N	D_f
Calibration in Figure 3.12 with standard-Box Counting	46	2.13
Calibration in Figure 3.13 with random-Box Counting	52	1.94

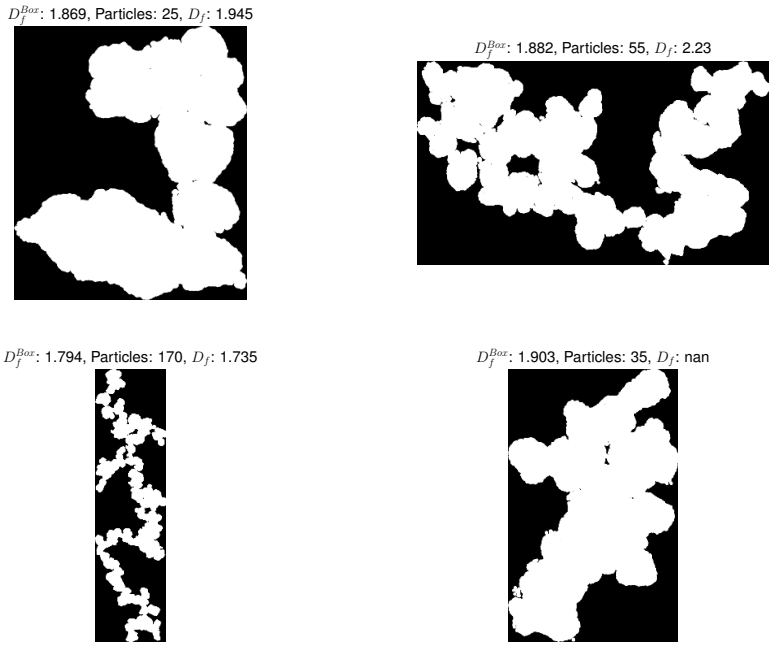

Figure 3.23: Four preprocessed real TEM images of soot aggregates. In the title of each, the Box Counting dimension, the number of counted particles and the estimated fractal dimension is shown. Here the calibration in Figure 3.12 have been used. nan is short for not a number, meaning that this combination of Box Counting dimension and number of particles is outside the valid area of the calibration.

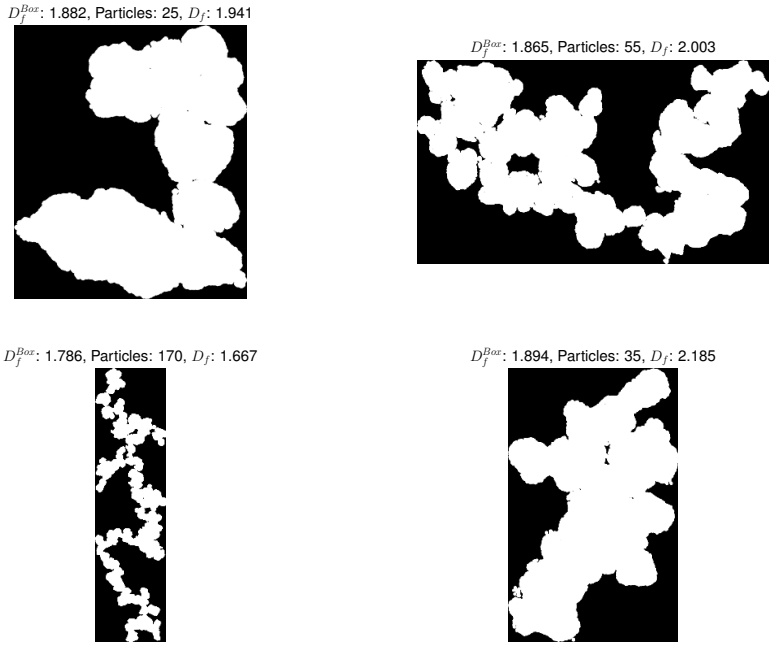

Figure 3.24: The same content as Figure 3.23 but with the random-Box Counting method and the calibration in Figure 3.13.

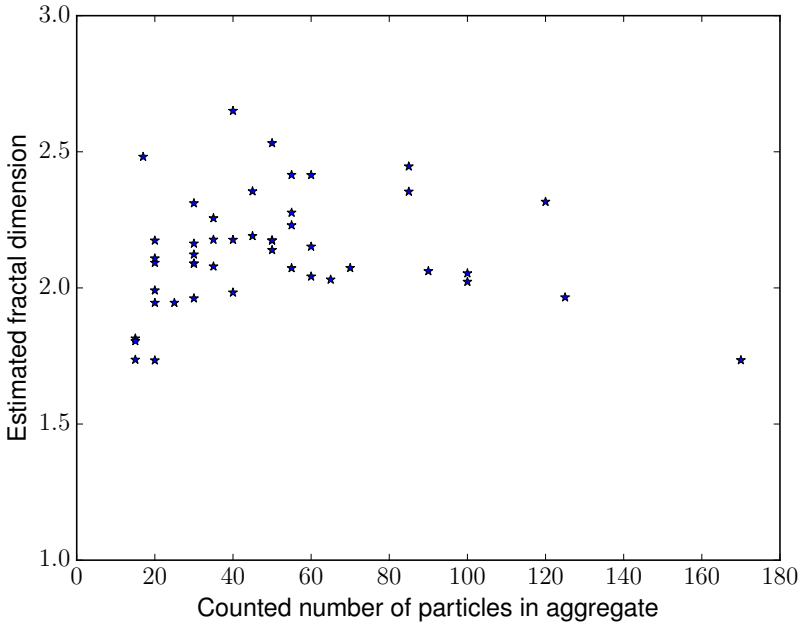

Figure 3.25: The counted number of particles in each aggregate plotted together with their estimated fractal dimension. These fractal dimensions have been estimated using the calibration from PC simulation and standard-Box Counting found in Figure 3.12.

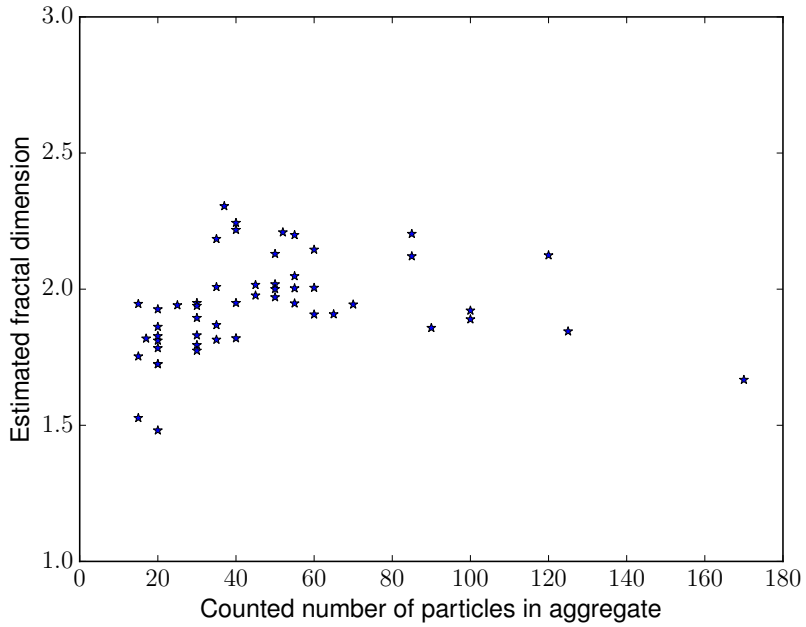

Figure 3.26: The same content as Figure 3.25 but with estimations using the random-Box Counting method.

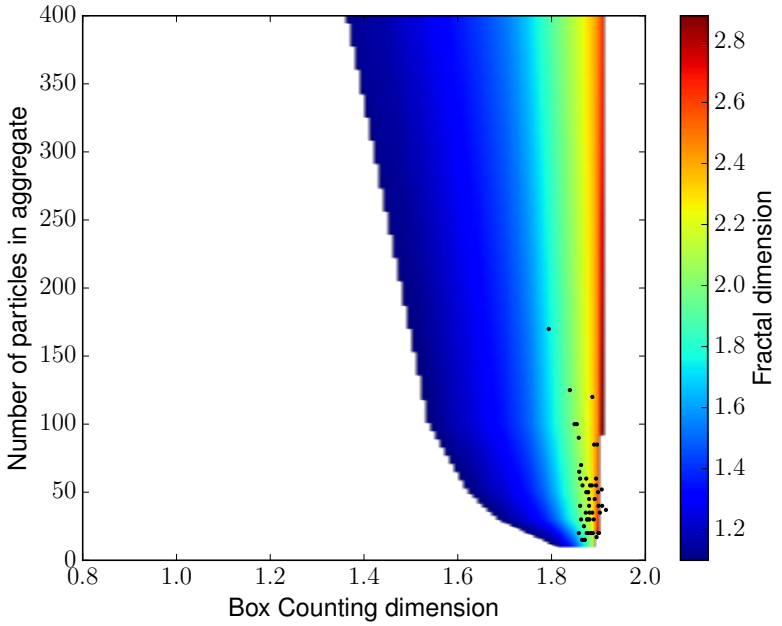

Figure 3.27: Visualisation of where standard-Box Counting dimensions and number of particles are found as points in the calibration. The calibration is the same as in Figure 3.12.

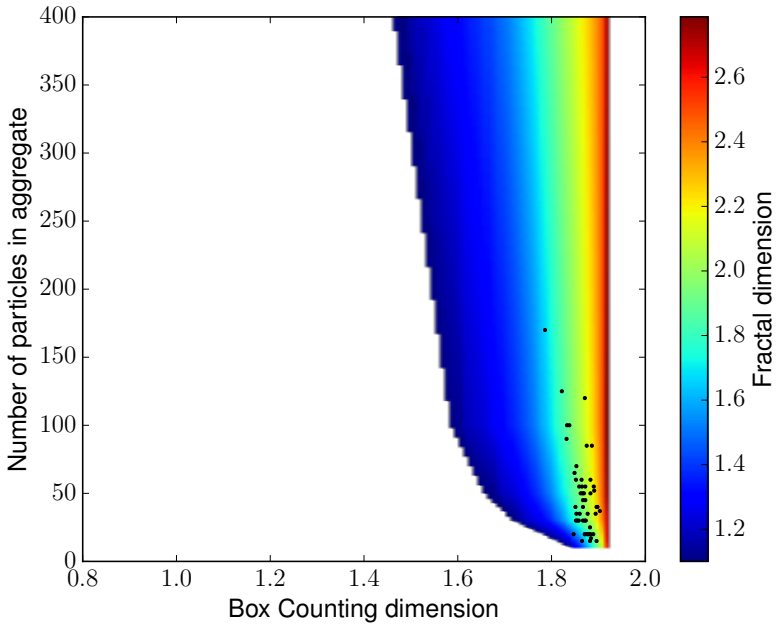

Figure 3.28: The same content as Figure 3.27 but with the calibration in Figure 3.13.

Chapter 4

Conclusions

The estimation of the fractal dimension from TEM images of soot aggregates is a problem where no obvious preferred method seems to exist. To use the Box Counting dimension as a feature in a machine learning approach is one of the newer ones. It has been interesting to implement and compare it to previous results of fractal dimension estimations on TEM images of soot aggregates.

The flowchart in Figure 1.1 show a visualisation of the connections between all data and methods in this master's thesis. It can be seen that everything is a reproduction and melding of previously developed methods. The following are the three main conclusions of using the machine learning approach with the Box Counting dimension for estimating the fractal dimension in TEM images. First of all, it will be close to impossible to develop an accurate method for estimating fractal dimension from projections with large Box Counting dimensions. This is connected to the loss of information in projecting a 3D aggregate into a 2D image. Secondly, a strong correlation between the Box Counting dimension and the fractal dimension can be seen in the calibrations. This verifies that the Box Counting dimension as a feature extracts the fractal properties from a TEM image which is promising for the future development of the method. Thirdly, two different Box Counting methods has been studied in this report and it is interesting to determine which one is most accurate. The standard-Box Counting method is connected to most of the negative attributes found in the discussion of the Box Counting methods. Examples of these attributes are: the facts for Wozniak et al. [5] to disregard it, the thin distribution of Box Counting dimensions for low fractal dimensions and the invalid calibration points. In the end the random-Box Counting calibration is the best one from the results in this report.

The random-Box Counting calibration and method has been applied to the real TEM images which results in one fractal dimension estimation for each image. The mean value of all these fractal dimension estimations is 1.94. This is a large value compared to corresponding estimations of previous works in Table 2.1, which could be connected to the difference in sampling methods. The soot in the TEM images of this project are sampled from the cold gas exhaust of a soot generator whereas the previous works sampled soot from inside a flame. There is a possibility that the soot aggregates, during the transport and cooling through the exhaust of the generator, increase their fractal dimension by folding themselves. This would mean that the soot aggregates in the TEM images used in this master's thesis actually have a larger fractal dimension compared to the ones used in previous studies.

To further improve the accuracy and understanding of this combination of methods, it is interesting to focus improvement and development on the following:

- Tune the parameters and possibly change the pipeline of the preprocessing algorithm to improve its performance and see how it affects the Box Counting dimensions.
- Expand the preprocessing algorithm to include a program counting particles in an aggregate.
- Explore the possibility of using the greyscale in a TEM image as a feature of the third dimension.

Appendix A

Calculating angle for a general triangle where all sides are known

$$\angle ACB = a\cos\left(\frac{|\mathbf{CA}|^2 + |\mathbf{CB}|^2 - |\mathbf{AB}|^2}{2|\mathbf{CA}||\mathbf{CB}|}\right). \tag{A.1}$$

Appendix B

Method for creating a vector with a certain tilt and rotation in relation to another vector This method concerns the 3D points A, B and C where only A is unknown. The following restrictions are set for the unknown point:

- The vector **CA** have the length r_A.
- The angle $\angle ACB$ is α.
- The Rotation for **CA** around the vector **CB** is β.

A can be calculated with the following algorithm.

1. Convert the point $(B - C)$ to spherical coordinates (r_B, θ, ϕ).
2. Create a point with spherical coordinates (r_A, α, β) and convert it to cartesian coordinates. Call this point $\hat{A}$
3. Perform two rotations on this point. First around the y-axis with angle θ and then around the z-axis with angle ϕ. This is performed with the following matrix multiplications:

$$\hat{A}_{rot} = \begin{pmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \hat{A}. \tag{B.1}$$

4. Translate the point to C as origin and

$$A = \hat{A}_{rot} + C. \tag{B.2}$$

Bibliography

[1] B.B. Mandelbrot and R. Pignoni. *The fractal geometry of nature*, volume 173. WH freeman New York, 1983.

[2] H. Omidvarborna, A. Kumar, and D.S. Kim. *Recent studies on soot modeling for diesel combustion.* Renewable and Sustainable Energy Reviews, 48:635 – 647, 2015.

[3] J. Hansen and L. Nazarenko. *Soot climate forcing via snow and ice albedos.* Proceedings of the National Academy of Sciences of the United States of America, 101:423–428, 2004.

[4] A. Seaton, D. Godden, W. MacNee, and K. Donaldson. *Particulate air pollution and acute health effects.* The Lancet, 345:176 – 178, 1995.

[5] M. Wozniak, F.R.A. Onofri, S. Barbosa, J. Yon, and J. Mroczka. *Comparison of methods to derive morphological parameters of multi-fractal samples of particle aggregates from TEM images.* Journal of Aerosol Science, 47:12 – 26, 2012.

[6] R.A. Dobbins and C.M. Megaridis. *Morphological Description of Flame-Generated Materials.* Combustion Science and Technology, 71:95–109, 5 1990.

[7] R. Thouy and R. Jullien. *A cluster-cluster aggregation model with tunable fractal dimension.* Journal of Physics A (Mathematical and General), 27:2953 – 2963, 1994.

[8] K. Skorupski, J. Mroczka, T. Wriedt, and N. Riefler. *A fast and accurate implementation of tunable algorithms used for generation of fractal-like aggregate models.* Physica A: Statistical Mechanics and its Applications, 404:106 – 117, 2014.

[9] Central Microscopy Research Facility, University of Iowa. *Transmission Electron Microsopy.* https://cmrf.research.uiowa.edu/transmission-electron-microscopy. Accessed: 2017-09-26.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, second edition, 2004.

[11] Ü.Ö. Köylü, G.M. Faeth, T.L. Farias, and M.G. Carvalho. *Fractal and projected structure properties of soot aggregates.* Combustion and Flame, 100:621–633, 1995.

[12] B. Jähne. *Digital Image Processing.* Springer-Verlag Berglin Heidelberg, 2005.

[13] K.B. Petersen and M.S. Pedersen. *The Matrix Cookbook.* Equation 84, 2005. URL `https://www.ics.uci.edu/~welling/teaching/KernelsICS273B/MatrixCookBook.pdf`.

[14] Satya Mallick. *Filling holes in an image using OpenCV*, 2015. URL `https://www.learnopencv.com/filling-holes-in-an-image-using-opencv-python-c/`.

[15] D. Eberly. *Minimum-Area Rectangle Containing a Set of Points.* 2015. URL `https://www.geometrictools.com/Documentation/MinimumAreaRectangle.pdf`.