

COMPARING DIFFERENT IMAGE PROCESSING APPROACHES IN THEIR ABILITY TO EXTRACT TWO DIMENSIONAL BUBBLE DISTRIBUTION

submitted by
Leon Baumann

to obtain the academic degree
Bachelor of Science (B.Sc.)

Friedrich-Alexander-University Erlangen-Nuremberg
Departement of Fluid Mechanics
Cauerstraße 4, 91058 Erlangen

Date of submission:
18.01.2021

Supervision:
Prof. habil. Dr.- Ing Antonio Delgado
Dr.- Ing. Bernhard Gatternig
Tobias Beck, M.Sc.

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, 18.01.2021

Abstract

Measurements of bubble size distribution (BSD) are pervasive in chemical engineering applications. For this task image processing constitutes a robust and widespread tool published in chemical engineering journals and other related fields. This thesis presents different image processing approaches for detecting bubbles in two dimensional images of multiphase bubbly flow. These methods include Hough Transform, which enables the detection of circular bubbles by voting on parameters accumulated in a transform space. Based on the topographic features of a gray scale image, the Watershed Transform segments individual bubbles using morphological operations. The last method separates overlapping bubbles by finding intersecting points (Concave points) based on convexity characteristics of the bubble contour. The three methods are compared in their ability to extract the two-dimensional bubble size distribution. Subsequently an algorithm, based on recent literature, is implemented for every method. For evaluating each algorithm two data sets are created: a synthetic data set, composed of 50 images with randomly generated bubbles of a given parameter distribution, and a real data set from an experimental setup, consisting of 30 manually segmented images. The segmented real data and the generated synthetic data is used as a ground truth for each segmentation algorithm. Probability density function (PDF) of bubble equivalent diameters, the Sauter Mean Diameter (SDM), and the f1-score were chosen to evaluate the performance of the algorithms. Correct segmentation of an algorithm is indicated by a Jaccard Similarity Coefficient above 0.5. For sufficient segmentation performance and an accurate bubble size distribution extraction, the f1-score of the respective algorithm on a given data set has to be at least 90 percent within a five percent significance level. A main finding of the thesis is that, following the given approach, the Concavepoint algorithm achieves the best results for both data sets.

Contents

Abstract	ii
1. Introduction	1
1.1. Intrusive methods	1
1.2. Non intrusive methods	2
1.3. Objective	3
2. Image processing basics	4
2.1. Digital images	4
2.2. Connectivity	5
2.3. Image noise	6
2.4. Spatial filtering	6
2.5. Morphological operations	7
2.6. Intensity thresholding	9
3. Material and methods	11
3.1. Software specifications	11
3.2. Performance measures	11
3.2.1. Probability density function	11
3.2.2. Sauter mean diameter	12
3.2.3. Jaccard similarity coefficient	12
3.2.4. Precision, recall, f1-score	13
3.2.5. Hypothesis testing	14
3.3. Data acquisition	14
3.3.1. Real data set	14
3.3.2. Synthetic dataset	16
3.4. Preprocessing	18
3.5. Algorithms	21
3.5.1. Hough transform	21
3.5.2. Watershed transform	24

Contents

3.5.3. Concavepoint algorithm	26
3.6. Parameter selection	29
4. Results and discussion	30
4.1. Manual segmentation of camera images	30
4.2. Synthetic bubble images	32
4.3. Parameter selection	33
4.3.1. Hough	33
4.3.2. Watershed	35
4.3.3. Concavepoint	35
4.4. Synthetic data set	38
4.5. Real dataset	39
4.5.1. Hough	39
4.5.2. Watershed	42
4.5.3. Concavepoint	44
5. Conclusion	47
A. General Addenda	48
A.1. List of materials used	48
List of Figures	49
List of Tables	52
Bibliography	53

1. Introduction

Gas-liquid contactors, especially bubble columns, are very common in many industrial fields, like the chemical, biochemical, or petrochemical industry. They are used in many applications, such as Fischer-Tropsch process for hydrocarbon synthesis, hydrogenation of unsaturated oil, coal liquefaction, fermentation, or waste water treatment [LDK13]. Especially bubble columns are popular because of their uncomplicated design and good heat and mass transfer properties. The rate of mass transfer is often the limiting factor in these reactions [Laa+05]. It is dependant on the liquid-side mass transfer coefficient k_L and the interfacial area a between liquid and gas phase [Fer+12]. Studies [Kul07; CM61] have shown that both k_L and a are related to the bubble size. In order to improve reactor design or to develop scale-up processes it is important to be able to measure the bubble size distribution (BSD) [Zho+16].

Different methods exist for measuring bubbly flow. Indirect methods deviate the bubble size distribution from physical parameters through mathematical descriptions [Kum+76; Yia+88]. However, they are not accurate for inhomogeneous flow fields, which occur in most technical applications [Maa+11]. Direct Methods can be further grouped into intrusive (see section 1.1) and non intrusive methods (see section 1.2).

1.1. Intrusive methods

Popular methods employed in the literature are capillary suction probes [BG91], conductivity probes [Wan+18], fiber optic probes [ELR05; RL07], and Wire-mesh sensors [Pra08]. Intrusive methods are simple to use and robust. Furthermore in systems with opaque walls and a high gas hold-up they are often the only option for bubbly flow measurements. However, intrusive methods disturb the flow field during their measurement, because they are installed inside the two-phase mixture. Additionally, to obtain overall data, multiple measurements at different locations are necessary due to their limited spatial resolution. This is time extensive and hence prohibits live measurements [Bus+13; FL16; BDW02].

1.2. Non intrusive methods

Non intrusive methods include Laser doppler anemometry (LDA) [SF86; Mar+81], Phase doppler anemometry (PDA) [Tso+01], X-ray [SL18], gamma ray computed tomography [Bie+13], and image processing/analysis. Often they can be applied globally with a high spatial resolution and make live measurements possible. The main limitations of non intrusive methods are the high price and the requirement of low gas hold-up, or a transparent liquid phase.

Image processing methods are able to acquire accurate BSD measurements. The bubble information (area, parameter, centroid location) are extracted from the two-dimensional projected area of the bubble in the image. However detection is difficult since the appearance of bubbles varies considerably due to different lighting conditions causing contrast reversal and interreflections [Str+16]. Furthermore, due to bubble breakup and coalescence, the shape of bubbles is often deformed. Bubble overlap in the form of bubble clusters occurs when the void fraction exceeds 1-2 % [Bet+20] making it difficult to separate and reconstruct the individual bubbles. Moreover it is difficult to determine the exact control volume that is being measured, because only the two-dimensional area of the bubbles is captured [Bus+13]. Langlard et al. [Lan+18] suggests grouping the image processing methods into non-parametric techniques, like morphological watershed transform, and parametric techniques, like hough transform and other direct object fitting approaches.

Taboada et al. [Tab+06] use an improved hough transform algorithm to reconstruct the original bubbles in a bioreactor. Brás et al. [Brá+09] identifies drops with different sizes in a liquid-liquid chemical process using hough transform.

Watershed transform [Mey94; BL00] belongs to the most used methods for bubble segmentation. Lau et al. uses the watershed algorithm on dense bubbly flow to separate bubble clusters, making it possible to study higher void fractions. Karn et al. [Kar+15] treats out-of-focus and in-focus bubbles as two separate population and analyzes them using morphological operations and watershed algorithm.

Honkanen et al. [Hon+05] introduces an approach that detects the boundaries of overlapping elliptical bubbles and separates the perimeter based on connecting points, groups the resulting perimeter arches and fits ellipses to reconstruct the original bubble. Similar to Honkanen et al. other authors adapt the same basic principle of identifying overlapping points on a boundary curve, grouping the separated boundary pieces and fitting an ellipse [Bet+20; ZJL12; Lan+18; Zaf+15a; Far+12]. These overlapping points are often referred to as concavepoints or breakpoints. Hence, in the remaining thesis this type of approach is named

concavepoint approach.

Lastly, using tracing particles in the fluid, Particle image velocimetry [Gra97] and Particle tracking velocimetry (PTV) [Cer+18; Mar+10] are popular image processing methods used for the measurement of velocity patterns.

1.3. Objective

In the literature many different image processing methods for the task of extracting the bubble size distribution of two phase mixtures are discussed. This thesis seeks to find the most satisfactory method among three prevalent approaches: Hough Transform, Watershed, and Concavepoint. Therefore, an algorithm for each approach shall be implemented and subsequently evaluated. Evaluation has to be robust and should indicate whether the respective algorithm is suitable for this segmentation task. The word segmentation is used to describe the extraction of gas bubbles from an image. The acquired data from the algorithms is compared to a ground truth. This ground truth consists of manually segmented images of real bubbly flow, taken from an experimental setup, and synthetically created bubble images. For sufficient segmentation performance and an accurate BSD extraction, the f1-score of the respective algorithm on a given data set has to be at least 90 percent accurate within a five percent significance level.

2. Image processing basics

This chapter introduces different concepts from the image processing domain that have been utilized for the research in this thesis. The fundamental concept of a digital image, the relation of its components, and different operations are elaborated.

2.1. Digital images

The following section about digital images is based on information from Gonzalez and Woods [GW18]. Digital images are saved as two-dimensional grids of points, i.e. pixels. It consists of M rows and N columns, the resolution of the image. This coordinate grid is referred to as the spatial domain. A higher resolution can display more spatial detail and therefore less information is lost in the digitizing process of a camera image. A digital image can be defined by a function $f(x, y)$, where $x = 0, 1, 2, \dots, M - 1$ and $y = 0, 1, 2, \dots, N - 1$. Each pixel stores information about its tonal value, the intensity, and can be distinctively referenced by its respective location (x, y) in the spatial domain. The pixel in the origin of the image has the index $(0, 0)$.

The intensity can be either black or white in the case of a binary image, shades of gray for a gray-scale image, or color in a colored image. Hence, different amount of storage is necessary to save images. In binary images only two bits are needed per pixel. Zero denotes black and one white. Pixels in gray-scale images usually take up eight bits, one byte, of storage. This allows the storage of 256 distinct shades of gray. Color information is typically stored in three different channels, one for each of the primary colors red, green, and blue (RGB). The combinations of these three primary colors are able to display any color. Each channel is a separate two-dimensional pixel-grid with commonly eight bits of storage per pixel. In total 24 bits of storage is needed for every pixel in such images. An illustration of a gray-scale digital image can be seen in figure 2.1.

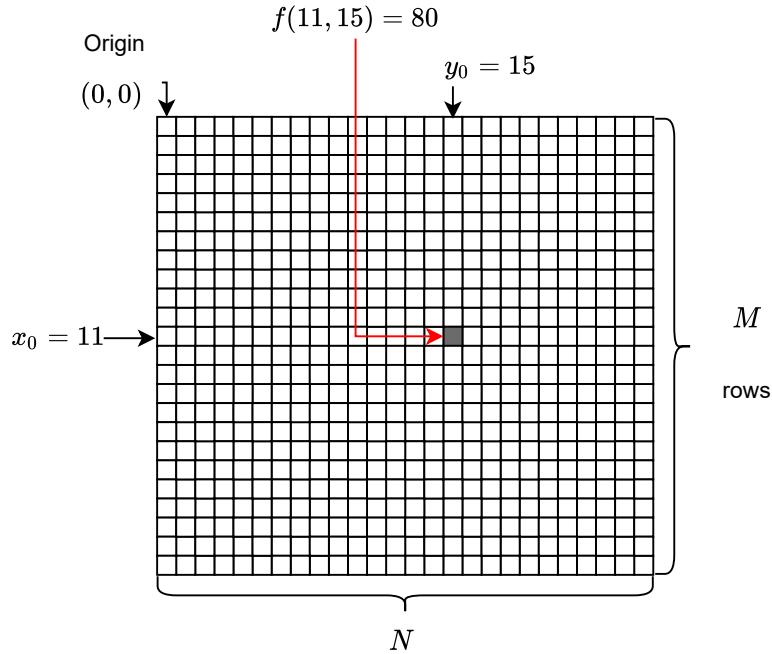


Figure 2.1.: Illustration of a digital image f . There is one gray pixel with the intensity of 80 at the position $x_0 = 11, y_0 = 15$.

2.2. Connectivity

The following information about connectivity is originated from Gonzalez and Woods [GW18]. There are two different types of connectivity between pixels within a certain neighborhood (see figure 2.2). The set of pixels, containing the two vertical and two horizontal neighbors of a pixel with the coordinates (x, y) is called 4-neighbors $N_4(x, y)$.

$$N_4(x, y) = \{(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\} \quad (2.1)$$

8-neighbors is the set if you add the four diagonal neighbors of the point.

$$N_8(x, y) = N_4 \cup \{(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)\} \quad (2.2)$$

Two points p and q have 4-connectivity if q is contained in the set $N_4(p)$ and 8-connectivity if q is contained in the set $N_8(p)$.

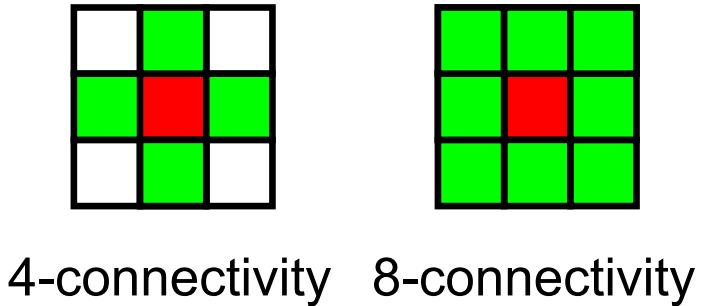


Figure 2.2.: Illustration of 4/8-connectivity. Connectivity is marked in green with respect to the center pixel in red.

2.3. Image noise

Noise in digital images is anything that is not part of the original signal actual scene that is captured and digitized. It mostly consists of random variations in pixel intensity values that occur during image acquisition, transmission or processing. Reasons are the statistical distribution of photons that are captured and the amount of electrons that are produced in the digitization process [Joh16]. It is common to assume that noise is independent of the spatial coordinates and the image scene itself. Different statistical models, like Gaussian, Gamma, or Rayleigh are used to describe the different kinds of noise distributions [GW18]. Noise during acquisition can be reduced by obtaining more information through capturing more signal, for example by using a longer exposure time. Noise in images, that has already been captured, can be decreased through image processing methods, like spatial filtering. However, noise is almost never fully removed and simultaneous altering of the original signal can not be avoided.

2.4. Spatial filtering

Spatial filtering is a method to reduce statistical noise in digital images. Information conveyed in this section is based on Gonzalez and Woods [GW18]. Each pixel value in an image is modified by a functions of the pixel and its surrounding values. It is assumed that a group of pixels with similar values represent a specific feature. Therefore modifying a single pixel value within a certain neighborhood does not necessarily decrease the information content of the captured object. The region/neighborhood that influences the pixel value is given by an array with varying size. These filtering arrays are further referenced as kernels. The

mathematical interaction between the kernel and the image is given by convolution (see equation 2.3). Convolution describes the process when a kernel ω is superimposed over every pixel (x,y) in an image f . Every kernel element is multiplied with the pixel underneath its position. The sum of all new elements gives $g(x,y)$, the new pixel value after the image was convoluted. The process of convolution is illustrated in figure 2.3.

$$g(x,y) = \omega \circledast f(x,y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s,t)f(x-s, y-t) \quad (2.3)$$

$a = (m - 1)$, $b = (n - 1)/2$ with n and m being the size of the kernel ω .

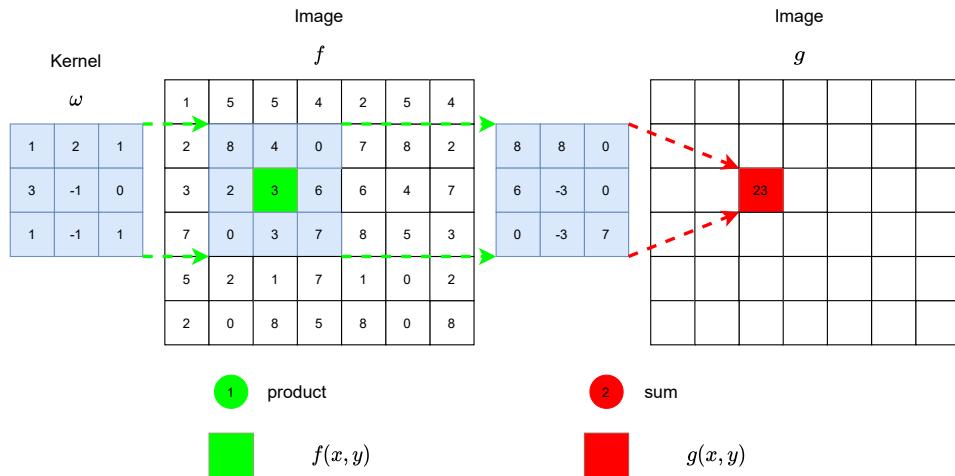


Figure 2.3.: Convolution of an image f with a kernel ω . The convolution is applied to the pixel in green. All pixel values underneath the blue kernel are multiplied with the respective kernel value. The sum of all products results in the new pixel value (marked in red).

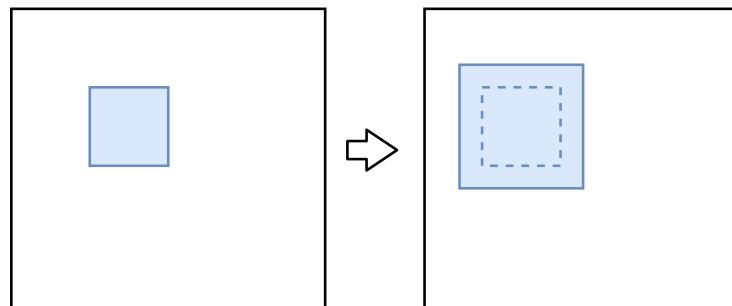
2.5. Morphological operations

Morphology in image processing is based on mathematical morphology. It is useful in manipulating binary images. The principle is that a structuring element S , similar to a kernel in spatial filtering, is slid over every pixel (x,y) in a binary image by translating its position. In a binary image usually white pixels represent foreground regions and black pixels background regions. A structuring element is a matrix with the size $m \times n$. Usually an odd size is chosen as it is convenient for calculations [GW18].

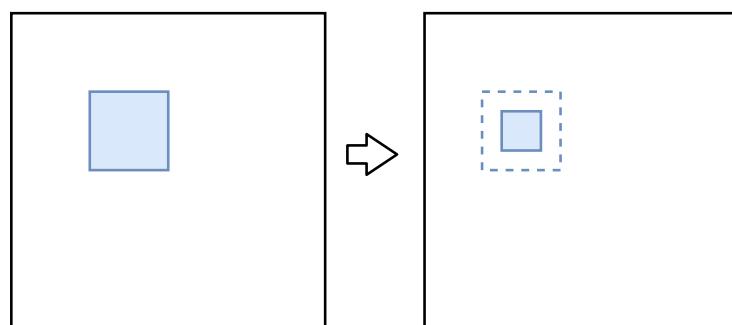
The first of the two basic operations in morphology is dilatation (see figure 2.4a). It is used to grow foreground regions and subsequently reduce holes inside these regions. It works

by sliding the center of the structuring element S over every background pixel in the binary image. At each pixel the structuring element is superimposed over the image. If at least one pixel intersects with the structuring element, the current pixel is set to the foreground value. However, if the image values beneath the structuring element correspond to the background value, the current pixel is left untouched [RW03].

The complementing operation to dilatation is erosion (see figure 2.4b). It has the opposite characteristics of dilatation. It shrinks foreground regions and enlarges holes within them. The center of the structuring element is superimposed over every foreground pixel. If there is any intersection with background pixels, the input foreground pixel is eroded away, by changing its value to the background value. Subsequently if no intersection with background pixels occur, the image is left as it is. The effect of both operations is heavily dependant on the size and type of the structuring element [RW03].



(a) Dilatation



(b) Erosion

Figure 2.4.: The left column illustrates a binary image with an object (blue square). The right column shows the object after dilatation (a) and erosion (b). Figure adapted from [GW18].

2.6. Intensity thresholding

Intensity thresholding is an image processing method for object segmentation. It works by converting a gray scale image into a binary image. It is assumed that all pixels, belonging to an object, share a similar pixel intensity. As mentioned before background regions are usually converted to black (0) and foreground regions, objects, to white (1). If all pixel values in an input image f are compared to a single threshold value T and subsequently replaced by either zero or one based on the condition in equation 2.4, it is referred to as global thresholding. $g(x, y)$ denotes the output pixel value [GW18, p. 743].

$$g(x, y) = \begin{cases} 0, & \text{if } f(x, y) > T \\ 1, & \text{if } f(x, y) \leq T \end{cases} \quad (2.4)$$

The global threshold value can be chosen based on the intensity histogram of the input image. It should separate two distinguishable peaks (bi-modal histogram), similar to figure 2.5. However, if multiple or no distinguishable peaks exist it is impossible for a single global

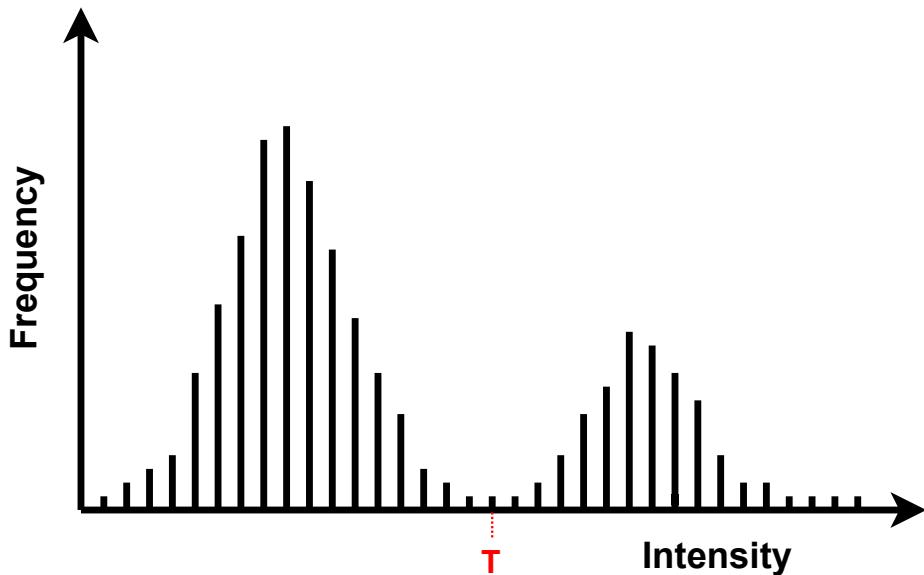


Figure 2.5.: Intensity histogram with a threshold value T marked in red. Figure adapted from [GW18].

threshold value to effectively segment objects and background. An approach that considers these local irregularities is adaptive/local thresholding. The general idea is that not single global threshold is determined, instead a local threshold is calculated dynamically within a certain neighborhood for every pixel in the input image. The underlying assumption is that a

smaller section of an image is more likely to depict a uniform intensity distribution. The local threshold for each pixel $T(x, y)$ is calculated with equations 2.5 , 2.6, and 2.7 (adapted from [GW18, p. 761]

$$T(x, y) = \bar{\omega} - c \quad (2.5)$$

$$\omega = (\omega_{x,y})_{m \times n} \quad \text{where } m, n = K_T \quad (2.6)$$

$$g(x, y) = \begin{cases} 0, & \text{if } f(x, y) > T(x, y) \\ 1, & \text{if } f(x, y) \leq T(x, y) \end{cases} \quad (2.7)$$

ω denotes the kernel matrix with the specified size K_T . c is a constant that is subtracted from the mean value of the kernel $\bar{\omega}$. This is necessary for areas with similar intensity, because the mean value is very close to the center pixel value. This ensures that the conversion is uniform.

3. Material and methods

3.1. Software specifications

All programming was done in a Python (Version: 3.8.3) environment. The image processing operations are implemented utilizing the OpenCV library (Version: 4.2.0.34) [Bra00]. OpenCV is an open source computer vision library offering a variety of pre-built functions and algorithms. However, different functions and classes are included from other open source libraries (scikit-learn (Version: 0.23.2) [Ped+11], scikit-image (Version: 0.17.2) [Van+14], Shapely (Version: 1.7.1) [Gil+–] , scipy (Version: 1.4.1) [Vir+20], numpy (Version: 1.18.5) [Har+20] , pandas (Version: 1.0.5) [McK10], matplotlib (Version: 3.2.1) [Hun07]). These additional libraries are utilized for image processing, evaluation, and data visualization.

3.2. Performance measures

In this section the different metrics for evaluating the image processing algorithms are described. Evaluation has to be robust and should indicate whether the respective algorithm is suitable for the segmentation task.

3.2.1. Probability density function

A number-based probability density function $P(d_{eq})$ using the equivalent diameter d_{eq} is employed to display the bubble size distribution.

$$d_{eq} = 2\sqrt{\frac{A_p}{\pi}} \quad (3.1)$$

A_p is the projected two dimensional area of each object[Men+05]. A gaussian kernel density estimation is used to estimate the probability density function from the frequency distribution.

$$\begin{aligned} P(d_{eq}) &= \frac{1}{n} \sum_{i=1}^n K\left(\frac{d_{eq} - d_{eq_i}}{h}\right) \\ K(d_{eq}, \sigma) &= \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d_{eq}^2}{2\sigma^2}} \\ h &= n^{-\frac{1}{d+4}} \end{aligned} \quad (3.2)$$

3. Material and methods

K is the standard 1D-gaussian Kernel. The bandwidth h is calculated by the Scott's rule with the number of datapoints n and the number of dimensions d [Sco15].

3.2.2. Sauter mean diameter

The second measure is the Sauter mean diameter (SMD). The SMD is used to characterize mass transfer in chemical reactions as it links the bubble surface area with its volume. The SMD can be calculated using formula 3.3 where n_i and $d_{eq,i}$ denote the number and equivalent diameter of a particular size fraction i .

$$SMD = \frac{\sum_{i=1}^k n_i d_{eq,i}^3}{\sum_{i=1}^k n_i d_{eq,i}^2} \quad (3.3)$$

A coinciding SMD value between two data sets indicates a similar bubble size distribution. [PMN98; KD16; Fer+12; Jun06].

However the SMD value does not consider the location of bubbles. It is possible for two images with the same size distribution but a different spatial distribution to have the identical SMD value. Therefore it is not sufficient to solely quantify segmentation performance using the SMD value.

In the evaluation the SMD values of the ground truth are compared to the SMD values of each algorithm. In order to determine how much they coincide the root mean square error RMSE is calculated [Hol17]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (3.4)$$

\hat{y}_i is the predicted value, y_i the ground truth value, and n ist the size of the data set.

3.2.3. Jaccard similarity coefficient

The Jaccard similarity coefficient (JSC) [CCT09] compares the similarity between two objects (see figure 3.1). It is calculated by correlating the area of a segmented object O_s with the corresponding ground truth object O_g according to formula 3.5 [Zaf+15b; Lan+18].

$$JSC = \frac{O_s \cap O_g}{O_s \cup O_g} \quad (3.5)$$

Values range from zero, no similarity at all, to one, two identical objects. The JSC value enables the grouping into true positive tp, false positive fp and false negative fn segmentation. A correct segmentation (true positive) is given when the JSC value is above a certain threshold T_{JSC} . T_{JSC} is set to 0.5 as recommended by different authors [Lan+18; Zaf+15a]. Contrary, if it

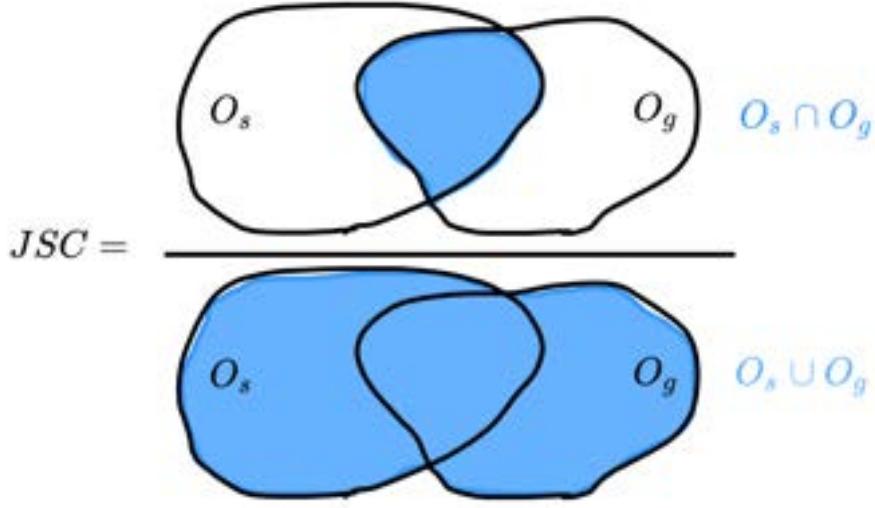


Figure 3.1.: Jaccard Similarity coefficient (JSC) calculation between a segmented object O_s and a ground truth object O_g . The area in blue represents the intersection and union respectively between the two objects.

the JSC value is below 0.5 the object is a wrong segmentation, a false positive. At last, false negatives are ground truth objects that have no corresponding segmented object with a JSC value above 0.5.

As a further metric for the overall segmentation performance the average of all JSC values of ground truth objects is calculated. This average is herein after referred to as the $AJSC$ value

3.2.4. Precision, recall, f1-score

Precision and recall (also referred to as Positive Predictive Value (PPV) and True Positive Rate (TPR)) are employed by many authors in the literature [Fat+10a; Zaf+15a; Lan+18]. Precision describes the fraction of correctly segmented objects among all detected objects.

$$precision = \frac{tp}{tp + fp} \quad (3.6)$$

Recall describes the fraction of correctly segmented objects among all ground truth objects.

$$recall = \frac{tp}{tp + fn} \quad (3.7)$$

The f1-score is the harmonic mean of precision and recall [Ilo+18].

$$f1 - score = \frac{2}{precision^{-1} + recall^{-1}} \quad (3.8)$$

3. Material and methods

It is suitable for quantifying segmentation performance, because it includes both over-segmentation (precision) and under-segmentation (recall) in a single value.

3.2.5. Hypothesis testing

Statistical hypothesis testing is used to assert whether an algorithm is suitable for the segmentation task. A left-tailed hypothesis test is used. By rejecting the null hypothesis H_0 : "The f1-score of algorithm x is not higher than 90 percent ($\mu \leq \mu_0 = 0.9$)."¹ it is concluded that the algorithm x is suitable. The sample size n is given by the amount of pictures in the data set. A significance level α of five percent, being a typical value for hypothesis testing, was chosen. The test statistic T is calculated as followed [OR18, p.273]:

$$T = \frac{\overline{\text{f1-score}} - \mu_0}{S} \sqrt{n} \quad (3.9)$$

$\overline{\text{f1-score}}$ is the mean f1-score and S the standard deviation of the sample. If T is bigger than the critical value z the null hypothesis is declined and subsequently the algorithm suitable for the segmentation. $z = 1.64$ is chosen from a table of quantiles of a normal distribution (see [OR18, p.319]).

3.3. Data acquisition

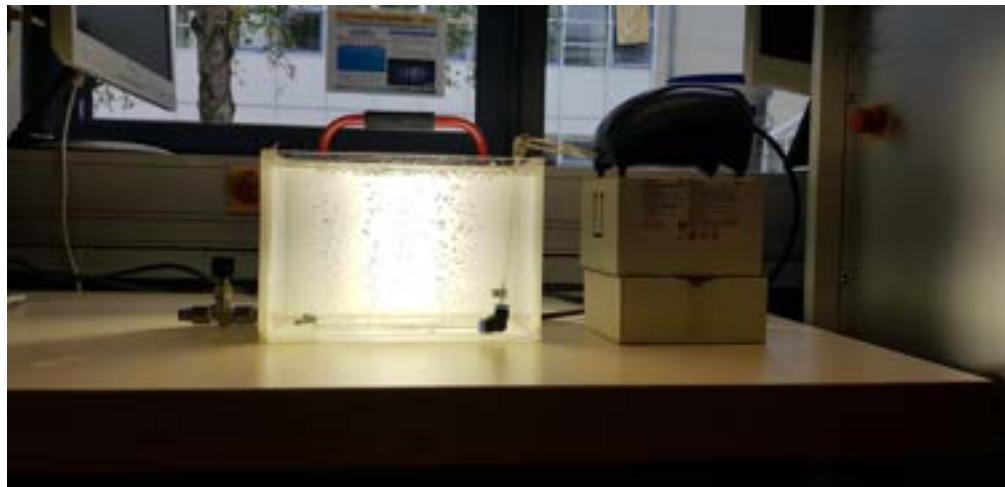
3.3.1. Real data set

This section describes the process of acquiring real images of bubbly flow, which serve as one of two data sets for the ensuing segmentation with the different image processing algorithms.

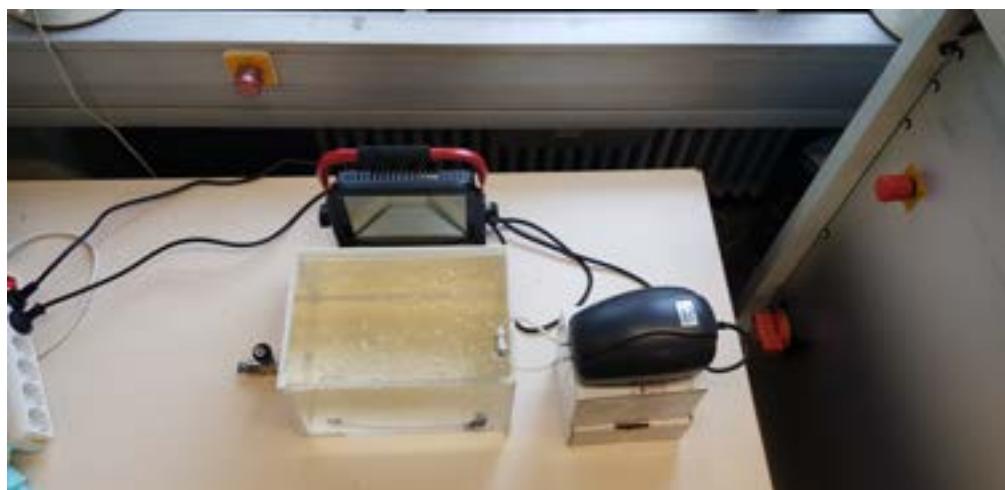
The main component is a 10 l transparent tank filled with water. Inside the tank, a punctured plastic tube with 37 holes is set up. The hole sizes vary in diameter in order to create bubbles in different sizes. To achieve overlapping bubble clusters holes are in close proximity to each other. The tube is connected to an air pump, PondoAir Set 450 (Pontec PfG GmbH, Hörstel), that induces the bubbly flow into the tank. A LED spotlight, Slimline Work Light (Luceco GmbH, Hamburg), is used to light the background. This is necessary to ensure a high contrast between background and gas bubbles for the subsequent image analysis. To attain a uniform background illumination a piece of white paper acts as a diffusor. A more comprehensive list of the material used can be found in the appendix A.1. Two pictures of the experimental setup are displayed in figure 3.2.

The camera, DMC-GX8 (Panasonic Deutschland GmbH, Hamburg), is fixed at 30 cm distance to the water tank. Pictures are saved with a resolution of 5200×3904 pixels. A focal

3. Material and methods



(a) Frontal view



(b) View from above

Figure 3.2.: Experimental set-up for the creation of the real data set.

3. Material and methods

length of 35 mm is chosen in order to display the section of the tank that contains the bubbly flow. The highest possible shutter speed 1/8000 s is selected to obtain a sharp image of the high velocity bubble stream. The highest aperture value was automatically set to $f/2.8$ by the camera considering the level of illumination. Generally speaking, a lower ISO value results in higher quality images, but due to the high shutter speed few light reaches the sensor and thus the images tend to be darker. Therefore a higher ISO value increases the contrast with the trade-off of elevating image noise. A ISO value of 400 was chosen. The camera settings for the experimental run were selected in order to receive sharp and high-contrast images. However, it is not in scope of this thesis to quantify the different camera settings regarding their influence on the performance of the image processing algorithms. Images from the experimental setup in their native size of 5200×3904 pixels are cropped to smaller sections with the arbitrary chosen resolution of 2000×1650 pixels to reduce computation time. The positions of the cropped sections are selected with a uniform random distribution across the original image as to not bias the selection process. Thirty images were chosen as an appropriate number for the manual labeling process as it is time intensive, but to still have a data set of considerable size.

The labeling process works by tracing each bubble outline with the mouse input. These points are then used to fit an ellipse using the direct least square method. This method from Fitzgibbon et al. [FPF99] is suggested by different authors [Hon+05; ZJL12]. The method minimizes the distance between data points and the fitted ellipse to produce the optimal set of ellipse parameters. Its advantages are the computational efficiency, the robustness to noise, and that it produces an ellipse result for all data input [MC04]. The ellipse fitting process has the advantage of simplifying storage of segmentation results, because an ellipse can be uniquely defined needing only five parameters (both center coordinates, the length of major- and minor-axis and the angle to the vertical axis). Whether this approximation of ellipse fitting is legitimate is discussed later. The data from the manual segmentation is written to a CSV-file and saved for the evaluation. Figure 3.3 depicts one of the acquired camera images in its original size, the cropped section, and the finished annotated image.

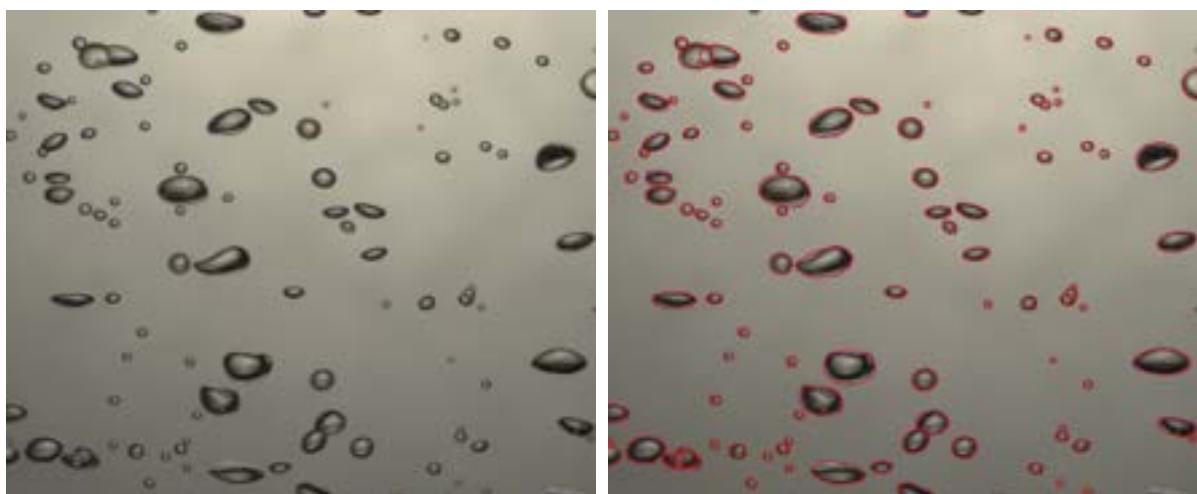
3.3.2. Synthetic dataset

To further investigate and quantify the uncertainty of the studied algorithms, synthetic images are created [Lau+13; Vil+19; FL16; Bet+20]. Ellipses are used to depict bubbles. Artificial images have the advantage that the exact amount and characteristics of all bubbles are perfectly known. This is not the case for the camera images of bubbly flow that were manually segmented in the previous subsection. The center location of each bubble is determined

3. Material and methods



(a) Original camera image with the size 5200×3904 pixels.



(b) Random cropped section with the size 2000×1650 pixels. (c) Result after the labeling process of the cropped section. Fitted ellipses are superimposed in red.

Figure 3.3.: Illustration of the labeling process for one of the camera images.

3. Material and methods

with a uniform distribution law and can be anywhere across the 2000×1650 pixels image. The angle to the vertical plane and the size of major- and minor-semiaxes are chosen based on observed data from the real data set. Slight variations in border thickness are induced depending on the size of the respective bubble. The axes lengths are also slightly varied with a standard deviation of 25 percent of the respective axis length. The border grey level intensity is set to 80. In addition, the border intensity is blurred with a 15×15 Gaussian kernel to accurately mimic the actual pixel intensity distribution along the border of bubbles taken with the camera. The background intensity was set to a fixed value of 200. The irregular illumination observed in the camera images is not considered in the creation of the synthetic images. One example of a synthetic image can be seen in figure 3.4.

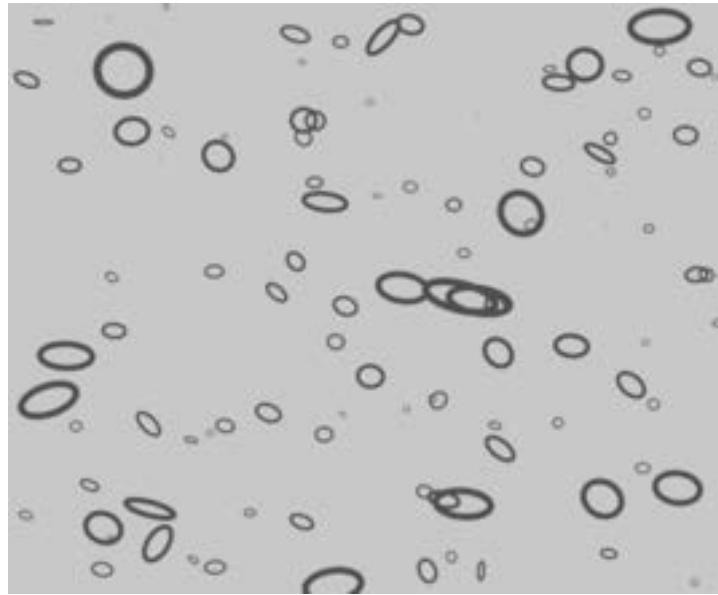


Figure 3.4.: Example of a synthetically generated image.

3.4. Preprocessing

This section details necessary steps in order to convert the input image into a binary image. The binary image functions as the direct input for all studied bubble detection algorithms. An overview of the preprocessing can be seen in figure 3.5.

The first preprocessing step is the grayscale conversion of the acquired images. Color information (chrominance) is removed, which leaves a single channel containing the graylevel intensity (luminance) ranging from 0 to 255. No relevant information is lost during this conversion as the camera image lacks color that would contribute to the segmentation process.

3. Material and methods

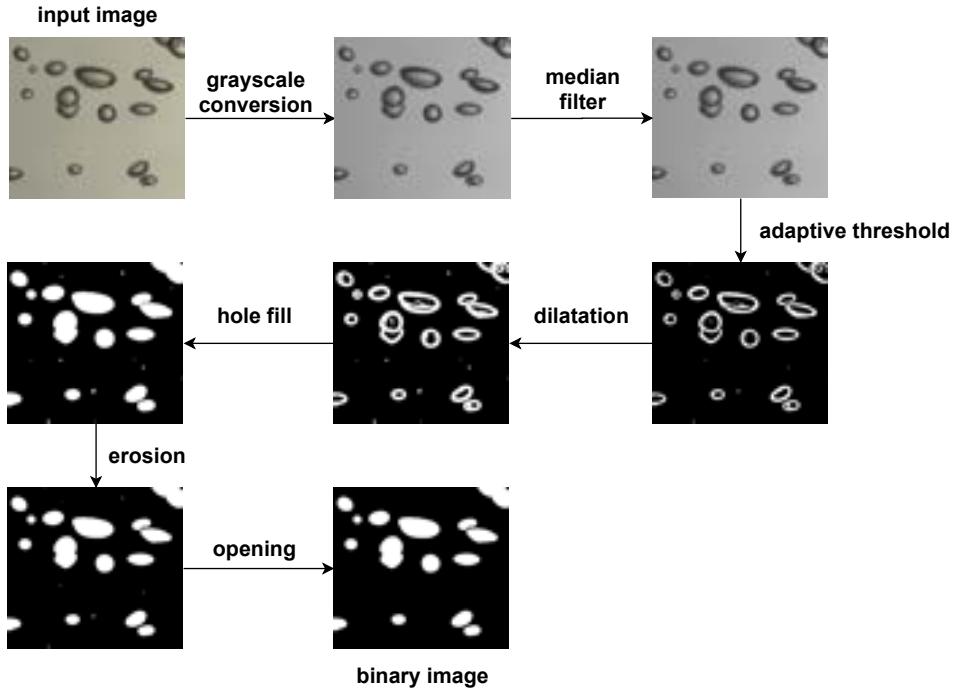


Figure 3.5.: Overview of the preprocessing procedure.

Furthermore, having a single pixel grid (containing only the graylevel intensity) reduces the complexity and run time of further image processing steps.

In order to reduce the spatially independent randomly distributed noise from image acquisition, spatial filtering is applied. Many authors in the literature apply median filtering [Hon+05; Zho+16; Tab+06; Lan+18; ASS]. It is a nonlinear filter that is able to preserve edges and reduces impulse noise [ZH12]. It ranks the pixels within a neighborhood, specified by the kernel size ($K_{\text{median}} \times K_{\text{median}}$), and the median value is selected as the new pixel intensity (see figure 3.6). Comparing the results of the different algorithms with different kernel sizes enables selecting the optimal configuration for noise removal.

After the spatial filtering has been applied and random noise considerably reduced the next step is to separate objects of interest, the bubbles, from the background. Zhong et al. [Zho+16], Langlard et al. [Lan+18], and Lau et al. [Lau+13] suggest using an adaptive thresholding method. This approach is advantageous because it considers uneven background illumination that is caused by light variations from the light source. Therefore an adaptive threshold as described in 2.6 is applied. The OpenCV function "adaptive threshold", based on the concept from Jaine [Jai89], is utilized.

After having applied the adaptive threshold, the image is converted to a binary image. However, due to the bubbles transparency, the inside of most bubbles has a similar intensity as the

3. Material and methods

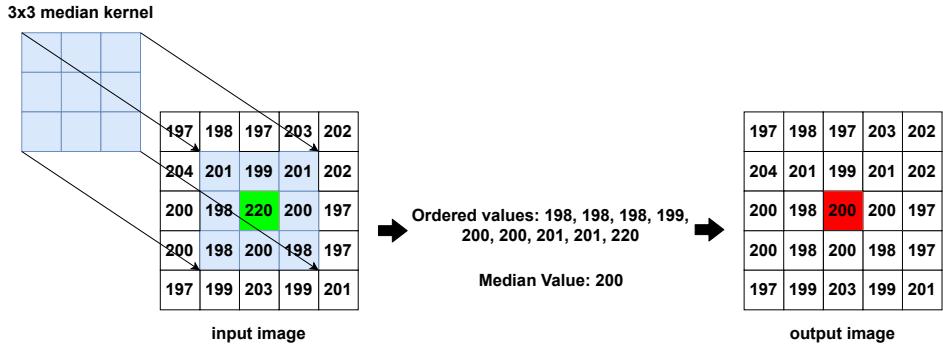


Figure 3.6.: Functionality of the median filter. The 3×3 median kernel is applied to the pixel highlighted in green. The median value within the specified neighborhood is selected and replaces the initial value. The new pixel value after the application of the median value is marked in red.

background and hence the thresholding process creates black background holes inside the bubbles. The next part illustrates the method to fill these holes.

Preceding the hole filling process, one iteration of dilatation is applied to close open contours. This is necessary, because the ensuing hole filling process does not fill open contours. A rectangle kernel is chosen. An example can be seen in figure 3.7.

The hole-filling process begins with the complement of the binary image being convoluted with a one pixel kernel using dilatation starting from the image border. The dilatation output is saved in an empty mask with the same dimensions as the input image. The dilatation continues until no further change is introduced by another iteration. The holes are not connected to the boundary and therefore remain unaffected. Lastly, the complement of the mask is taken which results in the binary image with the holes filled. However, in order to fill objects that are touching the image border, a white border is constructed around the image. This prevents the dilatation from invading the holes inside these objects. A one pixel big gap has to remain black, otherwise the hole picture would be recognised as a hole and subsequently be filled.

To counter the boundary enlarging from the dilatation one iteration of erosion is consecutively applied with the same kernel size.

The last preprocessing step is morphological opening. It consists of an erosion followed by a dilatation. Small noise objects that originate from the adaptive thresholding are removed. The effects of the opening operation can be seen in 3.8. Objects that only consist of a small number of pixels, and therefore do not represent any bubbles, are removed. Object size is determined based on 8-connectivity. The exemplary histogram with the opening operation

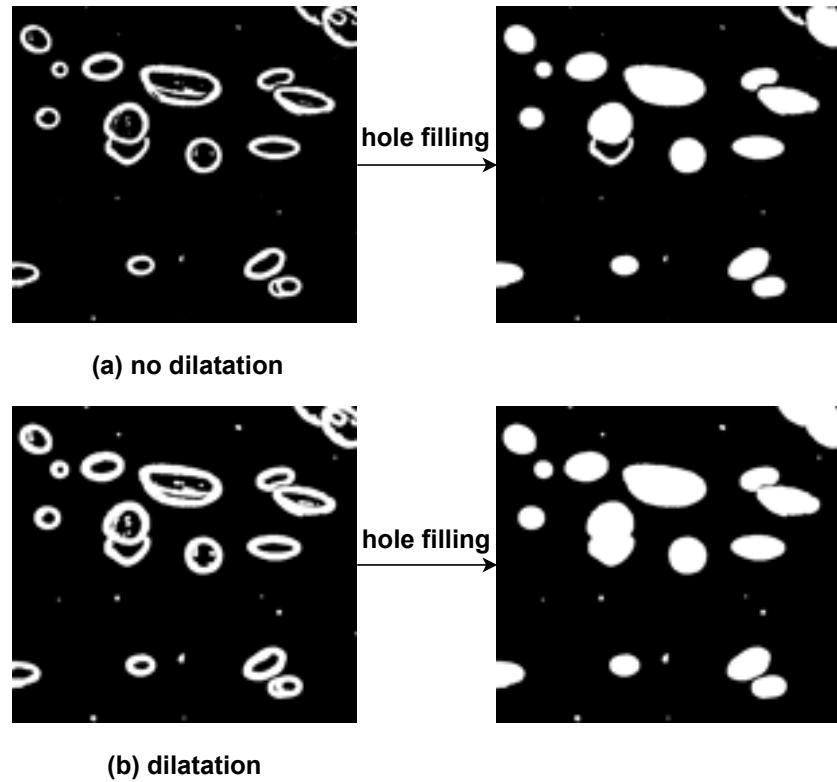


Figure 3.7.: Hole-filling process with and without prior dilatation.

(see figure 3.8a) shows no objects under a size of 20 pixels. According to the ground truth data from five manually segmented images there are no bubbles with a size below 24 pixels.

3.5. Algorithms

3.5.1. Hough transform

Hough Transform is a popular technique in computer vision used to detect shapes or features in digital images. Hough [Hou62] came up with the underlying principle while trying to detect tracks of subatomic particles. The original or standard Hough transform (SHT) developed by Duda and Hart [DH72] is able to detect lines in digital images. Compared to testing lines given by all point pair combination, the Hough transform is computationally more efficient. Furthermore Hough transform is robust against partial occlusion and shape defects of objects [Dav87; Har09]. The standard hough transform can only detect features that are expressible through a parametric form. Examples are lines, circles or ellipses. However different variants of Hough Transform have been developed over time. The Generalized

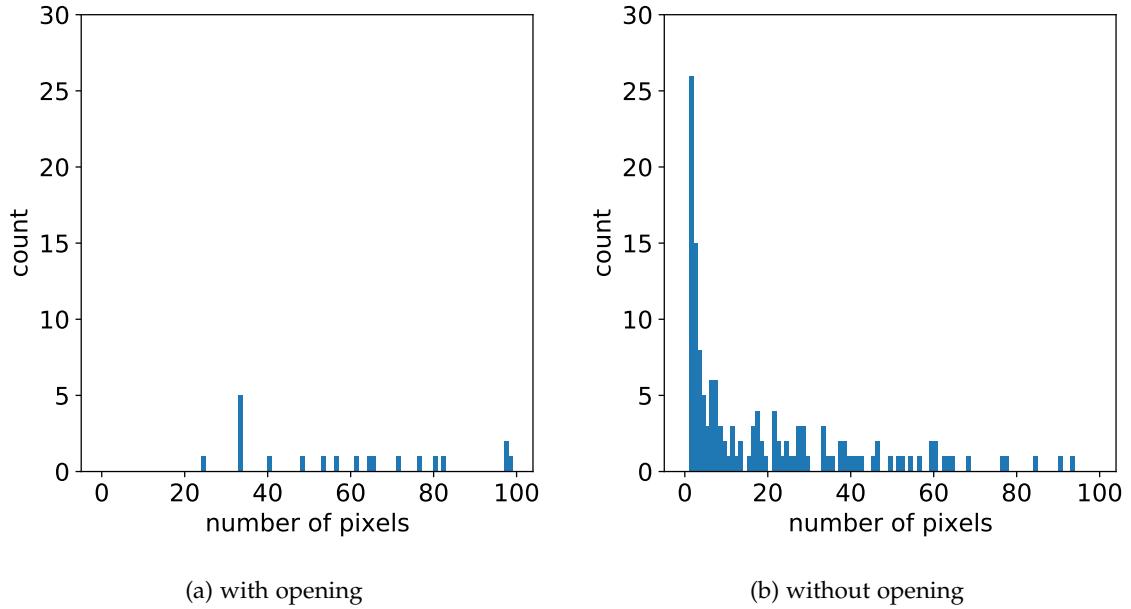


Figure 3.8.: Histogram of objects with a size between 0 and 100 pixels with and without the opening operation in advance.

Hough Transform (GHT) introduced by Ballard [Bal81] is able to detect non-parametric curves. Also probability based Hough Transforms like Probabilistic Hough Transform (PHT) or Randomized Hough Transform (RHT) have been developed. PHT takes not all object pixels as an input, but a limited number of randomly selected pixels, and therefore reduces the complexity significantly [MC15]. RHT, proposed by Xu et al. [XOK90], uses converging mapping instead of diverging mapping and dynamic storage in place of an accumulator array [XO09].

The basic principle of the standard hough transform for line detection is explained further. The input is a binary image. Any point (x, y) from the binary image could be part of a line or a set of lines. A straight line can be parameterized by the angle θ of its normal and its algebraic distance ρ to the origin. Each point in the $x - y$ image plane maps to a sinusoid in the $\theta - \rho$ transform plane. Intersections in the transform space and their respective ρ and θ parameters define the line in the original image plane. The intersections are counted inside an accumulator array. Peak values in this array are likely to represent lines in the image [Har09; DH72]. The proposed algorithm for bubble detection is based on the Hough Circle Transform. Kimme et al. [KBS75] adapted the SHT to be able to detect circles. This is useful and can be applied for detecting circular bubbles in the studied images. A circle is uniquely defined by

3. Material and methods

specifying the center coordinates (a, b) and the radius r . The parametric equation of a circle is [MC15]:

$$r^2 = (x - a)^2 + (y - b)^2 \quad (3.10)$$

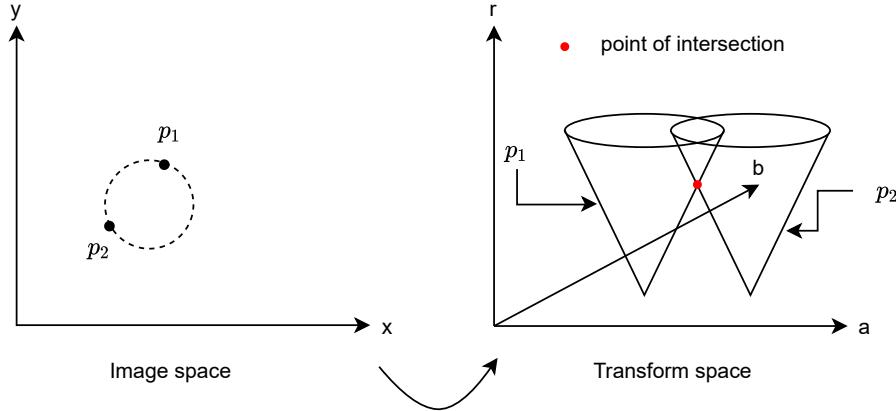


Figure 3.9.: A circle with two points p_1, p_2 in an image space visualized in the transform space.

The point of intersection between the two cones indicates the circle parameters.

Figure adapted from [Kie92].

Three parameters require a three-dimensional transform space. Every object point in the image plane creates a cone in the parameter space. Similar to the original Hough Transform a high number of intersections indicates the parameters of the studied feature, circles in the case of HCT, in the image plane. A visual illustration of the mapping from image space to transform space for the HCT can be seen in figure 3.9. OpenCV has a pre-build HCT function that is based on [KBS75] approach. This HCT function is applied on the binary image from the preprocessing.

The size of the accumulator array is set to one, so the resolution matches the original image. Furthermore the minimum distance between two neighboring circles, the maximum radius, and the minimum radius can be specified. Lastly it offers the functionality of selecting a accumulator threshold value. This threshold value decides which parameters in the transform space corresponds to real circles. The smaller it is, the more false circles are detected. The minimum radius is set to zero and the maximum radius to 100. Bubbles in the experimental setup do not exceed a radius of 100 in the data of five images. Setting a range of reasonable radii prevents the detection of circles that can not resemble a real bubble and furthermore speeds up the computation because a smaller range of parameters has to be counted. The values for the minimum distance $minDis$ between circles and the accumulator threshold value T_{acc} are varied in the ensuing parameter selection to find the best performing values.

3.5.2. Watershed transform

Watershed transform is another image processing approach for image segmentation. It can be grouped as an morphological image segmentation tool. The general idea is best visualized using a topographic flooding analogy. A topographic landscape is flooded with water from holes in the ground and basins begin to fill. Where different basins start to merge borders, i.e. watershed lines are created. For an input gray scale image the elevation in the topographic landscape is introduced by the pixel intensity. More specifically, the algorithm starts at seeding points, that are determined in advance. Sequential region growth is initiated in these seeding points. The growth advances in the direction of increasing gray scale intensity. In regions with constant intensity the growth happens in the same extend synchronously [MG98].

The original method proposed by [Beu79] uses local minima in the gray scale image as seeds to initiate the watershed transformation. Later the gradient image was utilized for the determination of seeding points [MB90]. However image noise, low contrast variations, or unclear boundaries prevent the determination of a single minima for each object [Att+16]. The many local intensity minima lead to heavy over-segmentation that limits the segmentation performance of the watershed algorithm [RM01]. Different approaches exist to redress this over-segmentation. Among these approaches are marker-based watershed [MB90; GB09; CR09], parallelization of marker-based watershed algorithms [MG98], or hierarchical watersheds [Beu94; Per+17]. The marker-based approach is implemented in this thesis. Markers are the seeding points for the watershed algorithm. A segmentation result with and without using markers can be seen in figure 3.12. Manually providing a set of markers for the segmentation is not a viable option in the bubbly flow measurement. A frequently used method for automatic marker selection is the distance transform. In the proposed algorithm euclidean distance transform specifically is employed [CYP04]. Euclidean distance transform calculates the nearest euclidean distance between every object pixel and the background. The euclidean distance between two points p_1, p_2 is defined as stated in equation 3.11 [CYP04].

$$d_{\text{Euclidean}}(p_1, p_2) = \|p_1 - p_2\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.11)$$

Figure 3.11 illustrates the euclidean distance of a square with the size 5×5 . The distance transform is applied on the binary image that was created in the preprocessing procedure. A graphic illustration can be seen in figure 3.10. The object points with the greatest distance to the background serve as markers [CR09]. They are selected by determining the local maxima, using 8-connectivity based connected-component analysis, in the distance map [MD02]. A threshold value minDis , the minimum distance between extreme points, influences the

3. Material and methods

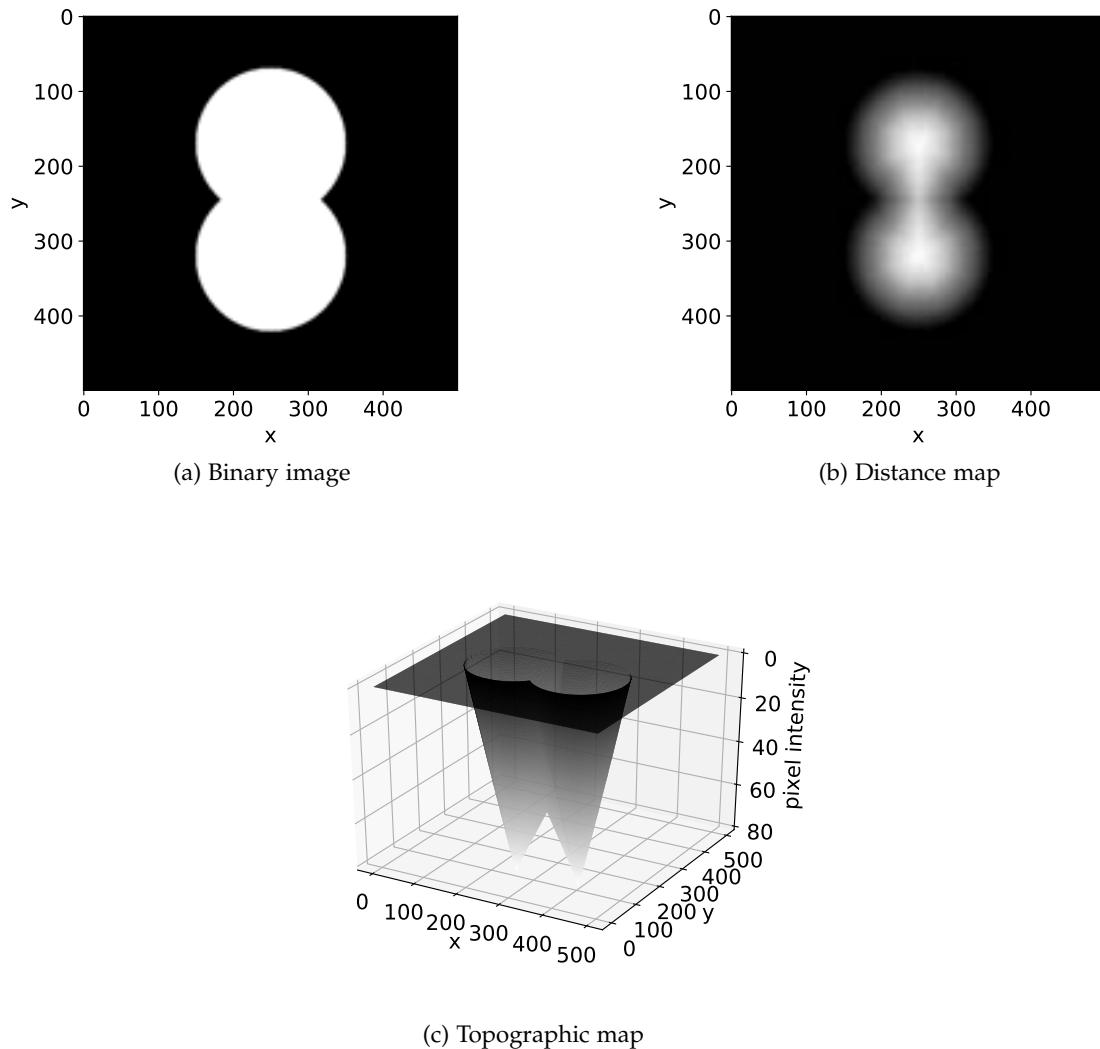


Figure 3.10.: A simple binary image of two overlapping circles with the corresponding distance map. The distance map is also visualized as a topographic map with the intensity as the z-axis.

$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
2	1	0	1	2
$\sqrt{5}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{5}$
$\sqrt{8}$	$\sqrt{5}$	2	$\sqrt{5}$	$\sqrt{8}$

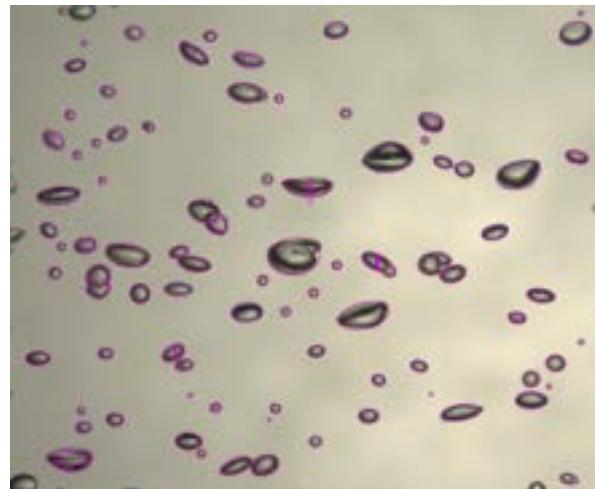
Figure 3.11.: Euclidean distance with respect to the center of the square.

extreme point selection. Accordingly, the watershed transform is applied on the negative distance map starting from the markers [MB90]. The complement of the distance map is taken because the watershed algorithm assumes that markers are local minimas. Lastly, the labeled regions, obtained by the watershed algorithm, are used to fit an ellipse with the direct least square method. This has the advantage that they can be easily stored in a CSV-file and compared to ground truth data.

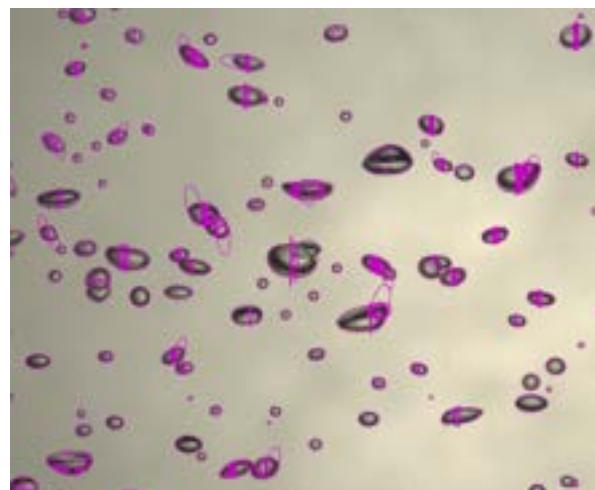
3.5.3. Concavepoint algorithm

Concavepoints/breakpoints are points on the contour of overlapping objects where object intersection occurs [Fat+10b]. They occur in regions with high concavity. The bubble segmentation is based on these concavepoints. Different methods exist in the literature to extract these points. Bettaieb et al. [Bet+20] use an angle feature and a linear correlation coefficient. Farhan et al. [Far+12] extract the concavepoints by connecting contour points and check if the line is inside the object. If the line is outside the object the concavepoint is determined by the maximum euclidean distance to the line. Zafari et al. [Zaf+15a] use a similar approach, also based on the definition of convexity. A polygonal approximation method is used for concavepoint detection by Zhang et al. [ZJL12]. Fu et al. [FL16] use the boundary curvature and image intensity gradient information. The proposed method finds the external contours in the binary image using OpenCVs implementation of Suzuki [Suz85] algorithm for border detection. Afterwards all contour points $p_c(x, y)$ are iterated and the vector to the next/previous point p_{c-i}/p_{c+i} , where i is the stepsize, determined. The angle θ between the three consecutive points is calculated by equation 3.12. An illustration can be

3. Material and methods



(a) with marker



(b) without marker

Figure 3.12.: Watershed with and without markers as seeding points. Heavy over-segmentation results without the selection of markers.

3. Material and methods

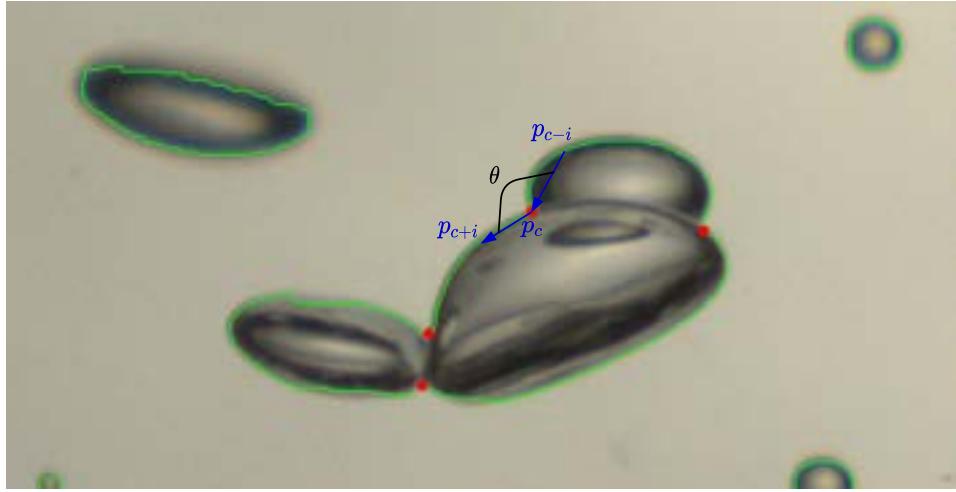


Figure 3.13.: Concavepoint determination process: The green lines are the detected contours from the binary image that are iterated to find concavepoints. Red points are the resulting four concavepoints.

seen in figure 3.13.

$$\theta(p_c) = \arccos \left(\frac{(p_c - p_{c-i}) \cdot (p_{c+1} - p_c)}{\|p_c - p_{c-i}\| \|p_{c+1} - p_c\|} \right) \quad (3.12)$$

To qualify as a valid concavepoint, two criteria have to be fulfilled. Firstly, the angle $\theta(p_c)$ has to be under a certain threshold T_θ . Secondly, the line $\overline{p_{c-i}p_{c+1}}$ must not intersect with the detected contour, i.e. has to be outside the detected object. After iterating through the entire contour, a number of possible concavepoints candidates are returned. However, to avoid having multiple concavepoints around the same object intersection, the concavepoint candidate with the highest T_θ in a certain neighborhood given by the distance threshold $minDis_{cp}$ is chosen. The concavepoints divide the contour into multiple pieces, referred to as perimeter arches. Each of these arches is used to fit an ellipse using the direct least square fitting method, described in 3.3.1. In order to avoid multiple ellipses representing the same bubble, which is the case with an even number of concavepoints, ellipses that are too close to each other are discarded and the concurring perimeter arches are combined to fit a single ellipse. The minimum distance between ellipse centers has to be under the threshold $minDis_{ellipses}$.

3.6. Parameter selection

This section describes the parameter selection for the three implemented algorithms. The ground truth data set, consisting of the thirty acquired camera images of bubbly flow, are used to select the parameters. The performance measure for the selection is the f1-score. A set of starting parameters is selected for each algorithm as a starting point for the parameter selection. The starting parameters ensure that at least a few bubbles are segmented correctly and hence a f1-score can be obtained. Otherwise the determination of the best performing parameters based on the evaluated output from the algorithms can not take place. Subsequently each parameter, starting with the pre-processing, is varied in the order they are applied in the algorithm. A reasonable interval is chosen for each parameter in order to save computation time and disk space. If the f1-score is a function of the respective value, the best value should be the global maximum. Thereupon the newly obtained parameter replaces the respective starting parameter. This is done until every parameter has the optimal value.

4. Results and discussion

This chapter asserts the quality of ground truth data, presents the selection of best performing parameters for each algorithm, and subsequently discusses each algorithms performance.

4.1. Manual segmentation of camera images

First the quality of the real data set is asserted. Figure 4.1 shows the probability density function of the real data set, consisting of the manual segmented camera images. A total of 3115 bubbles were marked across the 30 images.

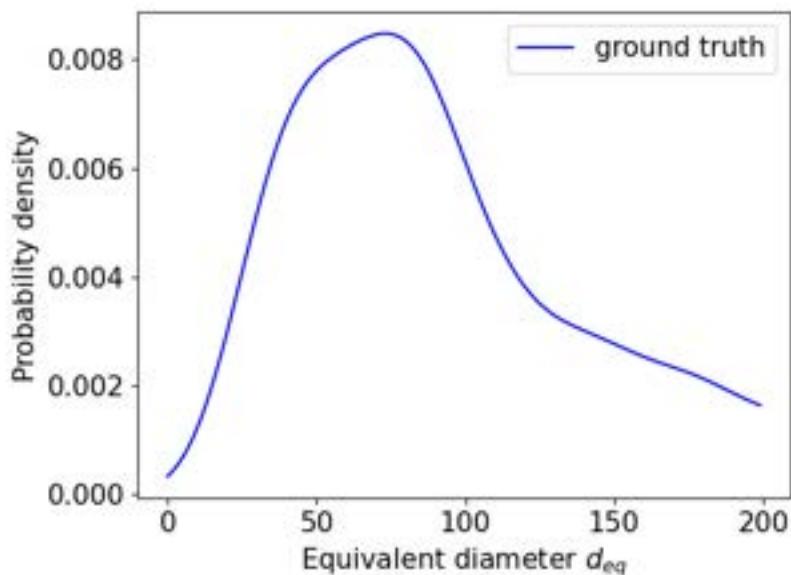


Figure 4.1.: Probability density function of the equivalent diameter for the real data set.

However, the marking process for the creation of a ground truth data set is not ideal and has some imprecision which needs to be considered.

At first it is possible that bubbles exist behind other opaque bubbles in the camera image and are therefore not visible. Furthermore some out-of-focus bubbles from behind the focal

4. Results and discussion

plane of the camera are indistinguishable from other noise sources, like surface scratches in the water tank. These reasons may cause bubbles to be missed during the manual marking of bubble contours. This underestimation means, independently of the algorithm, that the total number of false negatives in the camera images should be higher [Ilo+18] and therefore the recall value is slightly overestimated.

Further problematic is that the exact contours of the bubbles are not always clear. Sometimes only a small fraction of a bubble is visible at the image border which makes it impossible to mark the exact contour. Also due to border intensity variations it is not possible to identify the exact contour. Some common problems that occur in the camera images are illustrated in figure 4.2

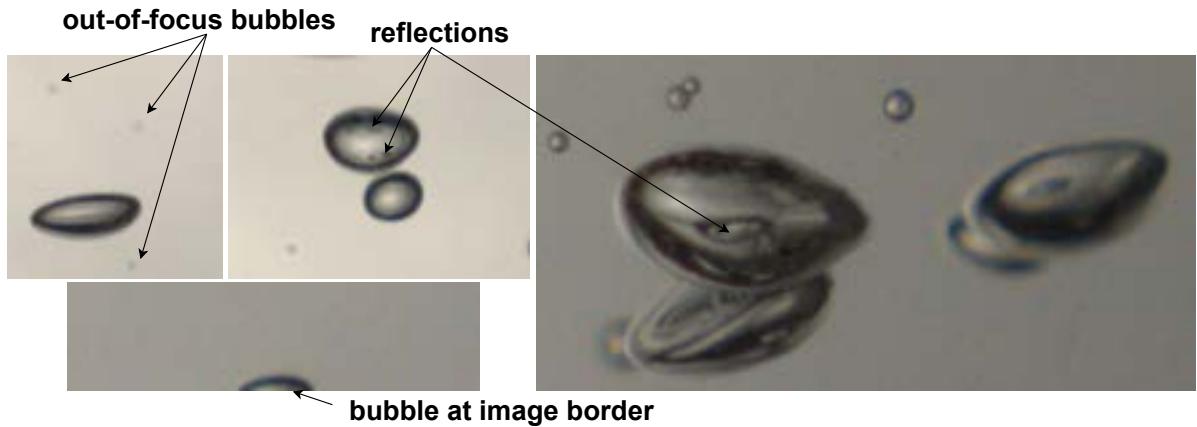


Figure 4.2.: Some visual artifacts in the camera images that prove problematic for the manual marking process.

As mentioned in 3.3.1 the contour of bubbles in the manual marking process are approximated by fitting an ellipse, making it easier to store segmentation results and to evaluate the algorithms. Furthermore, the ellipse fitting process manages to reduce the mouse input error caused by user imprecision. However, it should be evaluated whether this approximation is legitimate. Considering this, the difference between saving each bubble as a polygon of its contour points or fitting an ellipse for each marked bubble is evaluated. For this test the bubbles in one of the camera images are annotated and the AJSC value between both methods is determined. The JSC value as discussed earlier is a good metric to determine how much two geometrical figures resemble each other. An AJSC value of 0.908 is ascertained. Two things are important to keep in mind however. First there is a larger divergence between the polygon and ellipse approximation for small bubbles. Small bubbles consist only of few pixels and therefore a larger uncertainty is induced when approximating a round object with square

pixels through polygon approximation. For larger bubbles the JSC value is much closer to one indicating a good match between the two approximations. Secondly both methods are mere approximations and the exact uncertainty can not be evaluated, because the true bubble contours are not known. However, it is concluded that the AJSC value of 0.908 is enough to indicate a strong agreement between both approximation methods and therefore legitimates the ellipse approximation.

At last it should be noted that the operator errors induced while marking are not quantified in this thesis.

4.2. Synthetic bubble images

This section examines the quality of the synthetic data set. Figure 4.3 shows the probability density function of the synthetic data set. The total number of created bubbles is 5030 across the 50 synthetic images. The PDF of the synthetic data set resembles the PDF of the real data set (see figure 4.1) closely. This validates the applicability of the synthetic data set for the evaluation of the algorithms.

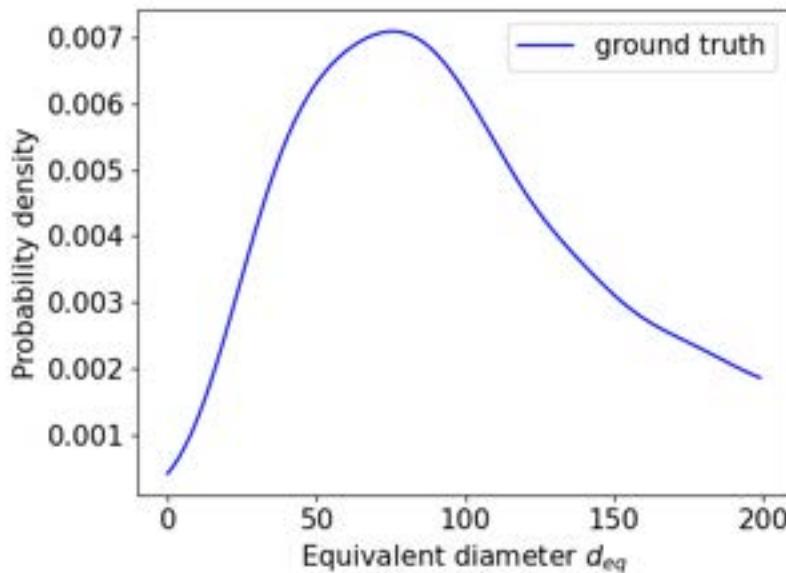


Figure 4.3.: Probability density function of the equivalent diameter for the synthetic data set.

As described in 3.3.2 the intensity of the ellipse contours has been blurred to resemble the original bubbles closer. Figure 4.4 shows the intensity distribution across the centerline of a bubble from a camera image and a synthetic ellipse depicting a bubble. Both bubbles have

roughly the same size. The main difference is that the bubble from the camera image has a smoother distribution curve with a lower gradient and the inside of the bubble has a slightly lower intensity than the background. Overall are the two intensity distributions very similar with respect to each other.

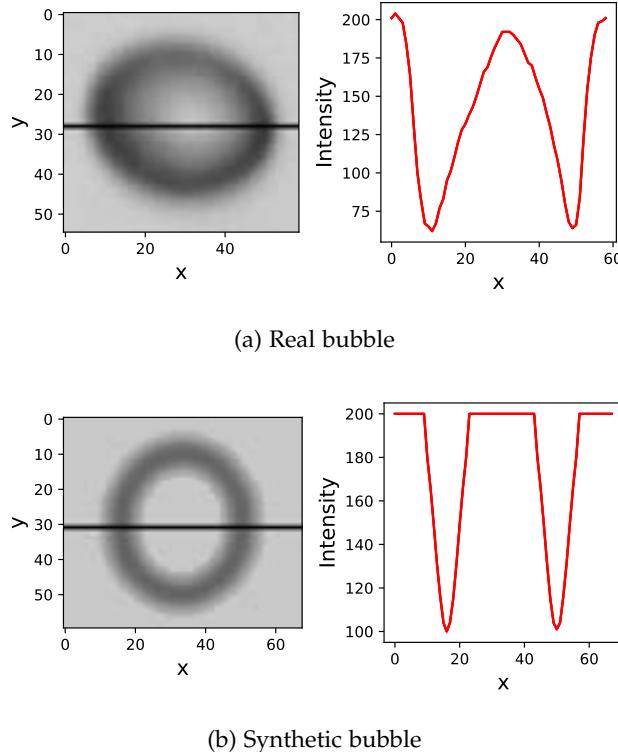


Figure 4.4.: Intensity distribution across the centerline (marked in black) of a real bubble (4.4a) and of a synthetic bubble (4.4b).

4.3. Parameter selection

This section discusses the results of the parameter selection for each of the algorithms. As presented in section 3.6 the value of each parameter is chosen based on highest f1-score.

4.3.1. Hough

Figure 4.5 shows the respective parameters of the Hough algorithm plotted against the f1-score. The amount of values that were tested for each parameter depended on whether a clear maximum was discernible. The best performing value is annotated in each graph.

4. Results and discussion

Table 4.1 lists the starting and final parameters, after the parameter selection for the Hough algorithm. The median value was varied in an interval of odd values from 0 to 19. When no median filter is applied the f1-score is zero. Near the end of the median interval the f1-score decreases. This can be attributed to the blurring of edges that occurs with bigger kernel sizes. The maximum performance was returned for a kernel size of five. The performance of local threshold kernel size K_T is zero outside the interval between 11 and 23. The performance maximum is at a value of 19 for the kernel. If the constant c is set to zero or one there is no output. The reason, as mentioned earlier, is that areas with a uniform intensity are converted wrong in the thresholding step. A value of three delivered best results for c . $K_{dil/ero}$ and K_{op} show a similar curve and are set to three and seven respectively. The minimum distance between circle centers $minDis$ has the best performance for a value of 36. Lastly the accumulator value T_{acc} has a clear maximum at 12.

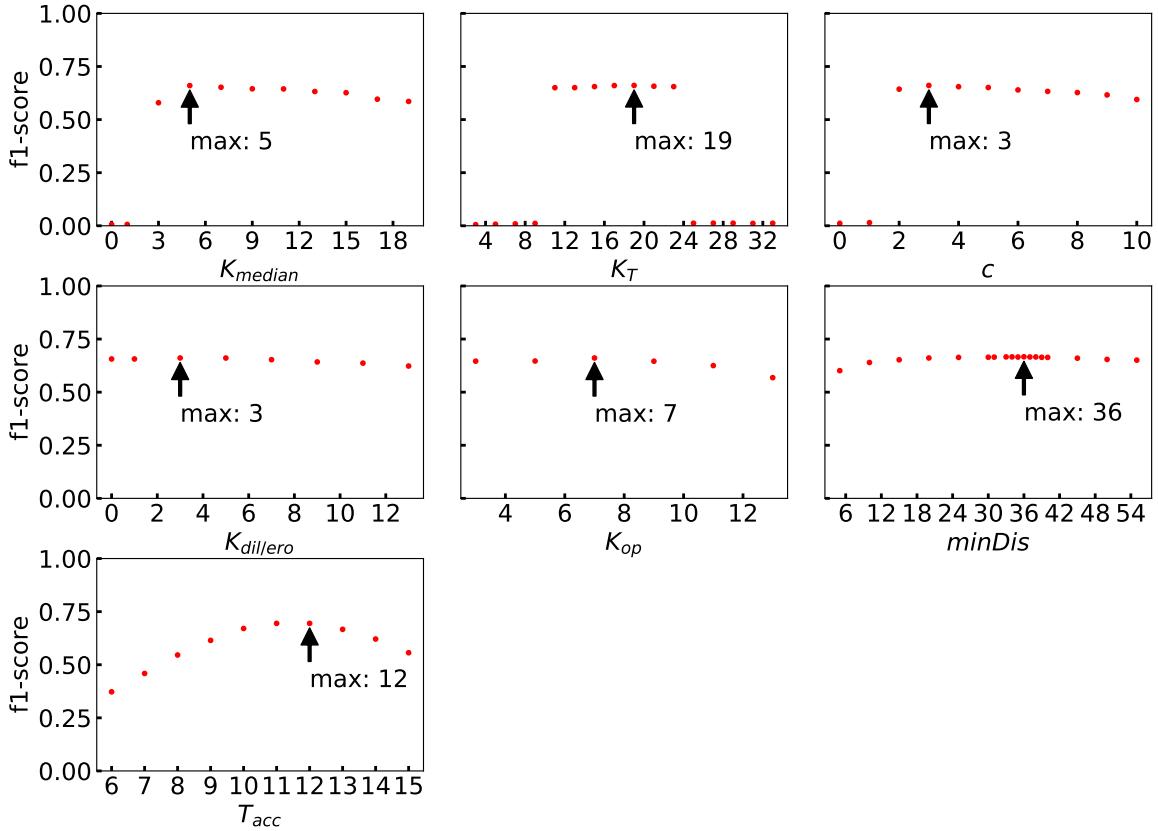


Figure 4.5.: Performance of the different parameters of the Hough algorithm. The best performing value is annotated for each parameter.

Table 4.1.: Parameters for the Hough algorithm.

	K_{median}	K_T	c	$K_{\text{dil/ero}}$	K_{op}	T_{center}	minDis
starting	5	17	3	5	7	20	13
final	5	19	3	3	7	36	12

4.3.2. Watershed

The same selection procedure is applied for the Watershed algorithm. Figure 4.6 depicts the different parameters of the watershed algorithm plotted against the f1-score. Table 4.2 lists the starting and final parameters, after the parameter selection for the Watershed algorithm. A kernel size of 11 delivered the best results for the median filter. The local threshold kernel size has the performance peak similar to the Hough algorithm at 17 and the constant c at three. $K_{\text{dil/ero}}$ and K_{op} have the highest performance, just as for the Hough algorithm, at five and seven respectively. The minDis between global minima in the distance map has the highest performance for a value of zero, so no distance restriction at all.

Table 4.2.: Parameters for the Watershed algorithm.

	K_{median}	K_T	c	$K_{\text{dil/ero}}$	K_{op}	minDis
starting	5	17	3	5	7	0
final	11	17	3	5	7	0

4.3.3. Concavepoint

At last the parameters for the Concavepoint algorithm are determined. Table 4.3 lists the starting and final parameters, after the parameter selection for the Concavepoint algorithm and figure 4.7 shows the different values tested for each parameter plotted against the f1-score. A median kernel size of five results in the highest f1-score. The optimal local threshold kernel size value is 13 and thereby slightly lower compared to the Hough and Watershed algorithm. According to its peak the constant is set to three. $K_{\text{dil/ero}}$ and K_{op} are set to seven and nine respectively. The best angle T_θ below which concavepoints are counted as valid arises to be 120. However, there is only a slight performance difference less than one percent for the values between 110 to 136. The *stepsize* has the maximum performance for a value of four. The minimum distance between concavepoints $\text{minDis}_{\text{cp}}$ and ellipse centers $\text{minDis}_{\text{ellipses}}$ only slightly impacts the performance (less than two percent). Their maxima are two and 30

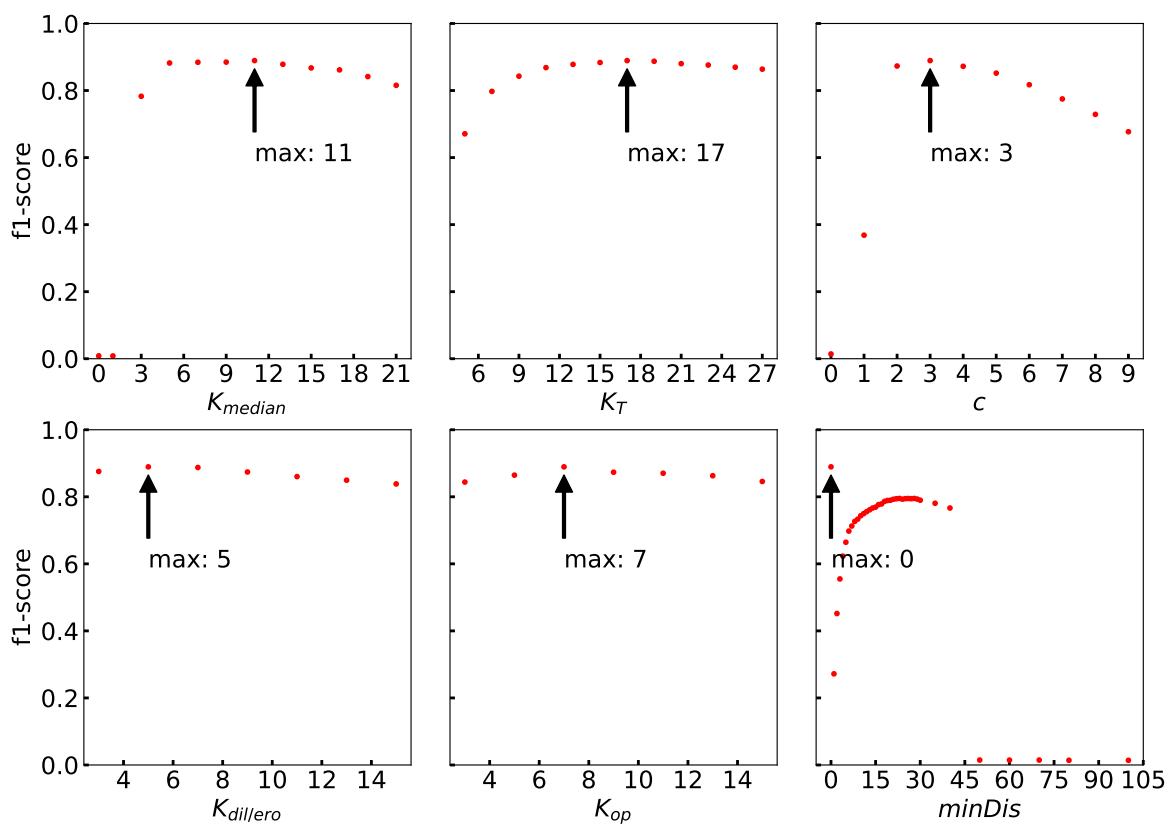


Figure 4.6.: Performance of the different parameters of the Watershed algorithm. The best performing value is annotated for each parameter.

respectively.

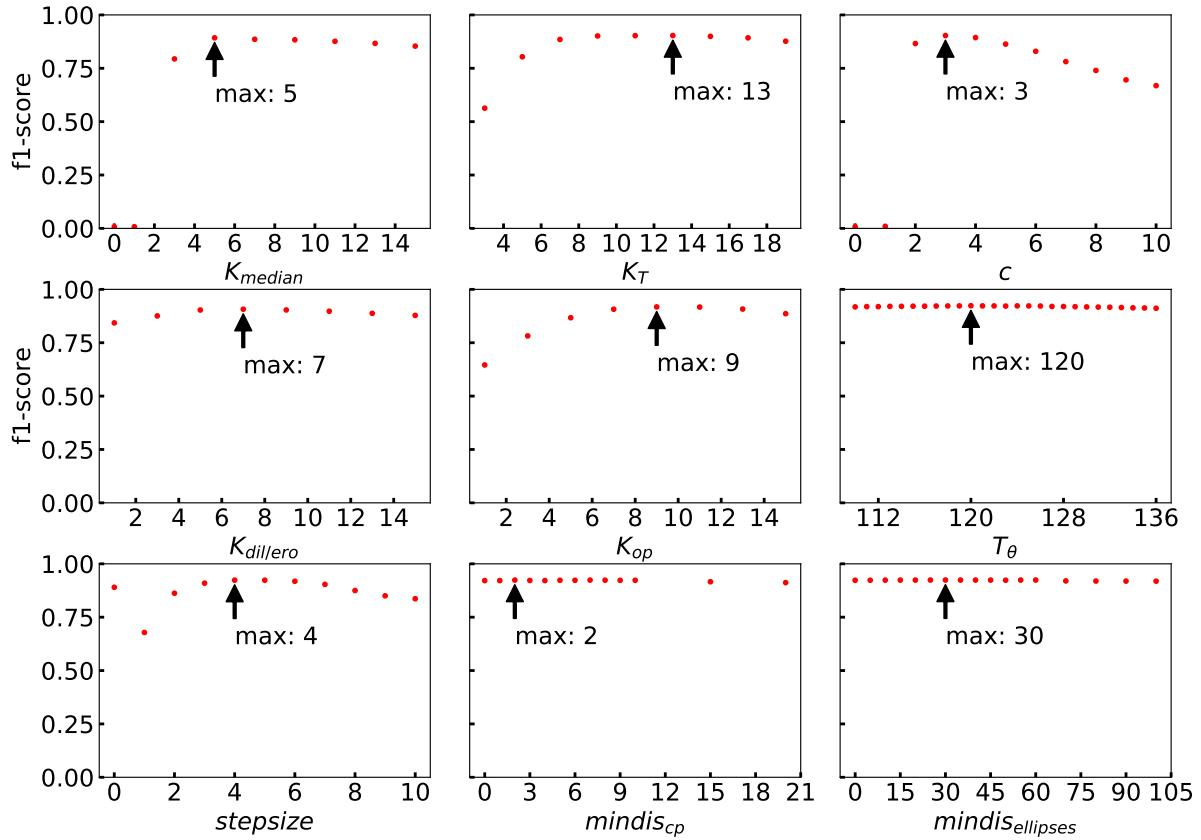


Figure 4.7.: Performance of the different parameters of the Concavepoint algorithm. The best performing value is annotated for each parameter.

The f1-score was selected for the parameter selection of all algorithms, because it considers over- as well as under segmentation. The Average jaccard similarity coefficient (AJSC) is not suitable. It is calculated by averaging all the JSC values of ground truth objects with their respective segmentation object (object with greatest overlap). However this does not consider over-segmentation, because segmented objects that have no overlap at all are not included in the calculation. Additionally the Sauter mean diameter is not reasonable, because it does not look at the position of different objects. Therefore it is possible for two segmentation results with a different spatial distribution of the segmented objects to have a similar SMD value.

Table 4.3.: Parameters for the Concavepoint algorithm.

	K_{median}	K_θ	c	$K_{\text{dil/ero}}$	K_{op}	T_{angle}	stepsize	$\text{minDis}_{\text{cp}}$	$\text{minDis}_{\text{ellipses}}$
starting	5	17	3	5	7	130	5	8	20
final	5	13	3	7	9	120	4	2	30

4.4. Synthetic data set

Performance of the algorithms is first evaluated utilizing the synthetic data set, consisting of fifty images of ellipses generated at random based on bubble data from the real data set as described in subsection 3.3.2. Each algorithm is applied to this set of synthetic images with the parameters selected in subsection 4.3. The performance of the algorithms is summarized in table 4.4. The recall values of all algorithms are, compared to the precision value, considerably lower. The ellipses in the synthetic image are created with a uniform random distribution. This makes it likely for smaller bubbles to be created within bigger bubbles. This bubble information is lost in the creation of the binary mask, explaining the lower recall value.

Table 4.4.: Performance of the the algorithms on the synthetic data set.

algorithm	f1-score	precision	recall	AJSC
hough	0.647	0.864	0.518	0.434
watershed	0.883	0.983	0.802	0.735
concavepoint	0.910	0.961	0.866	0.796

Figure 4.8 shows the probability density functions and SMD scatter plots of all three algorithms for the synthetic data set. It is visible that the PDF distributions of all three algorithms are shifted towards larger equivalent diameters, confirming the proposition that small enclosed bubbles are missed. Furthermore, The RMSE values are all above 20 suggesting that the agreement of SMD values between the ground truth and the algorithms is not high. The scatter plots of the Watershed and Concavepoint algorithm show a clear overestimation of the SMD values indicated by the higher y-intercept of the regression line, due to enclosed bubbles being missed and the lacking segmentation of some bubble clusters. This coincides with the distribution shift observation in the probability density functions. The Hough algorithm underestimates the SMD value for the majority of images in the data set, because it is unable to recognize large elliptical bubbles. Lastly according to the f1-score the Hough algorithm is performing the worst and the Concavepoint algorithm the best. This is

also confirmed by carrying out hypothesis testing to study whether one of the algorithms meets the hypothesis. The results are summarized in table 4.5. T is the test statistic and z the critical value as described in 3.2.5. The Concavepoint algorithm is the only algorithm among the three to exceed the critical value with a T value of 2.22. Therefore it can be concluded that the Concavepoint algorithm is able to accurately extract the BSD for the synthetic data set.

Apart from the lower recall value, a higher f1-score for the synthetic data set could be achieved by conducting a separate parameter selection for the synthetic data set. Although the general size distribution and appearance of ellipses in the synthetic data set is based on the real data set, ellipses remain an approximation for the bubble contours.

Table 4.5.: Test statistic T for all algorithms of the synthetic data set.

algorithm	T	z
hough	-33.05	
watershed	-3.12	1.64
concavepoint	2.22	

4.5. Real dataset

The main evaluation is carried out by applying the different algorithms on the real data set and investigating whether they are a suitable measurement method to study bubbly flow with respect to its size distribution according to the hypothesis in the introduction.

4.5.1. Hough

This subsection compares the output of the Hough algorithm on the real data set consisting of thirty acquired camera images. Table 4.6 lists the mean of each performance measure.

Table 4.6.: Performance of the Hough algorithm on the real data set.

algorithm	f1-score	precision	recall	AJSC
hough	0.695	0.816	0.606	0.490

The high precision value of the hough algorithm indicates that there are few wrong detection (false positives) among the segmentation result. This can also be seen in the example

4. Results and discussion

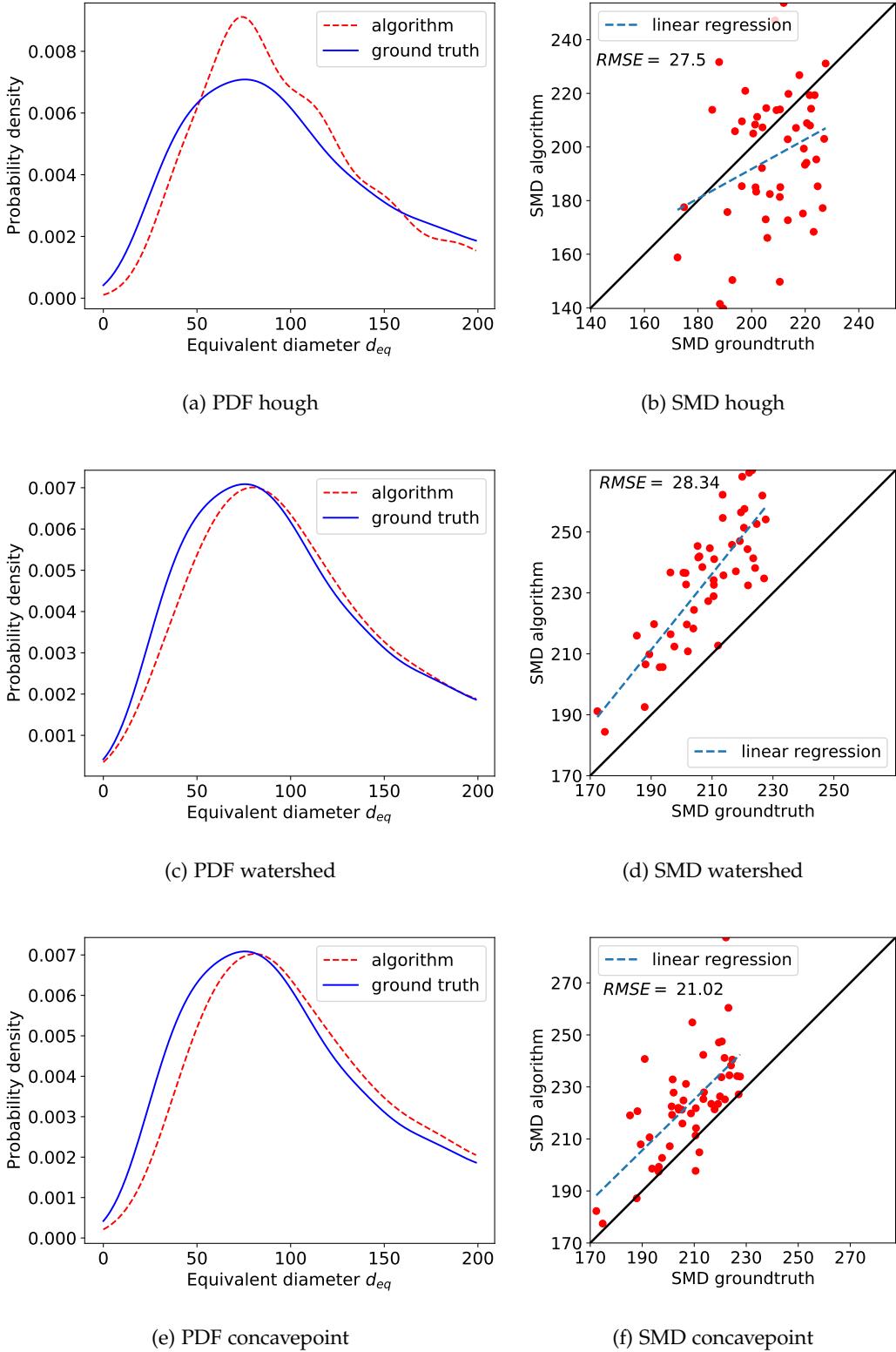


Figure 4.8.: Probability density function of the three algorithms and synthetic ground truth data. Scatter plot of the SMD values of the three algorithms and synthetic ground truth data.

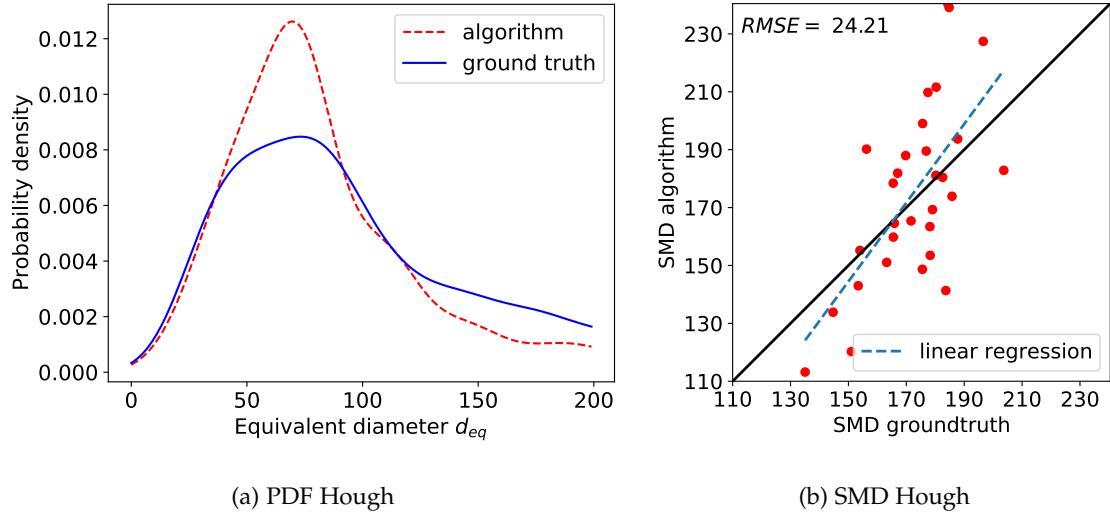


Figure 4.9.: Probability density function of the Hough algorithm and real ground truth data
 (a). Scatter plot of the SMD values of the Hough algorithm and real ground truth data (b).

segmentation in figure 4.10. However many bubbles are not recognized by the Hough algorithm leading to the low recall value. Furthermore the average area intersection (AJSC value) between ground truth objects and the segmentation result is not exceeding 50 percent. Reasons are the high number of false negatives and furthermore a circle is in cases of bigger deformed bubbles not a good fit for the bubble contour. The probability density function (see figure 4.9a) shows the average distribution of the equivalent diameter of the Hough algorithm and the ground truth data. Comparing the two distributions reveals a significant difference. The Hough algorithm detects more bubbles in the equivalent diameter range of approximately 45 to 90 pixels, but at the same time increasingly underestimates bubbles in the diameter range starting from 90 pixels. The reason for this behavior is that the Hough algorithm fails to resolve big deformed, mostly elongated, bubbles correctly. They are either not detected at all or wrongly segmented into multiple smaller bubbles. Figure 4.9b portrays a scatter plot of the ground truth SMD values plotted against the SMD values of the Hough algorithm for the thirty images respectively. The root mean square error RMSE of 24.21 does not indicate a good agreement between the ground truth and predicted data. The linear regression line intersects with the identity line. This indicates the SMD value of the Hough algorithm is both under- and overestimated for the data set. At last hypothesis testing, as described in 3.2.5, is applied to check whether it is correct to assume, based on the 30 camera images,

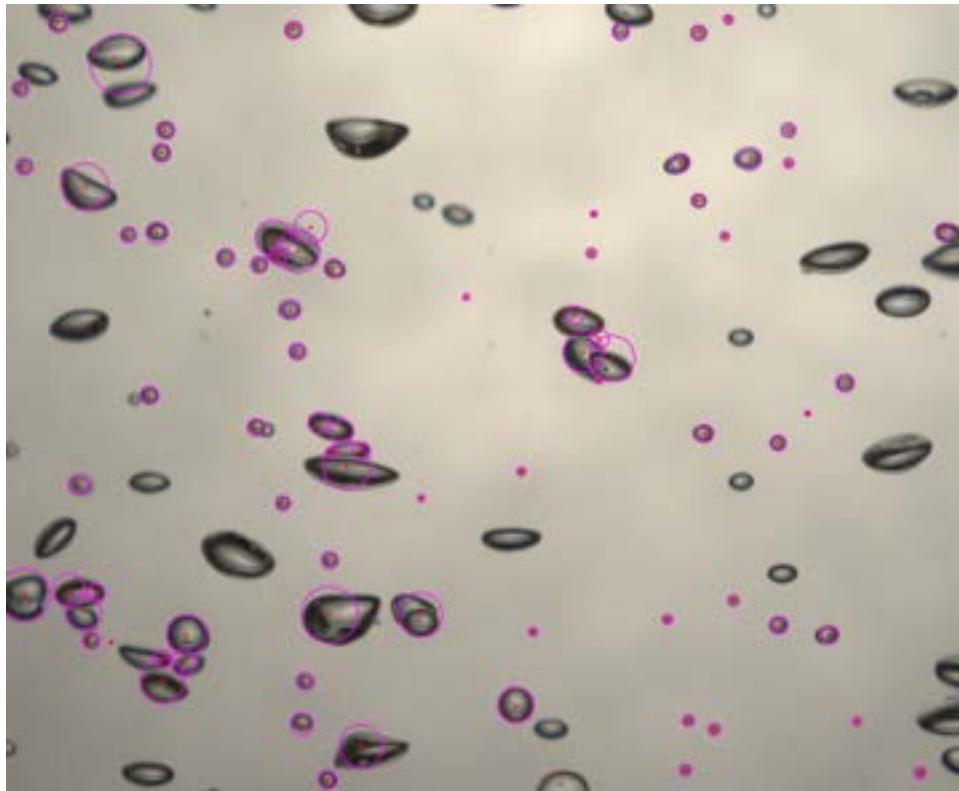


Figure 4.10.: Segmentation result of one of the camera images with the Hough algorithm.
Detected circles are marked in magenta.

that the Hough algorithm is able to extract the bubble size distribution accurately. With the f1-score of 0.695, a standard deviation of 0.047, and a sample size of $n = 30$ the test statistic T results in a value of -31.0. Because $T = -31.0 < z = 1.64$ the null hypothesis is accepted and it can be concluded that the Hough is not able to extract the BSD well from the camera images.

4.5.2. Watershed

This subsection compares the output of the Watershed algorithm on the real data set. Table 4.6 lists the mean of each performance measure.

Table 4.7.: Performance of the Watershed algorithm on the real data set.

algorithm	f1-score	precision	recall	AJSC
watershed	0.889	0.901	0.879	0.746

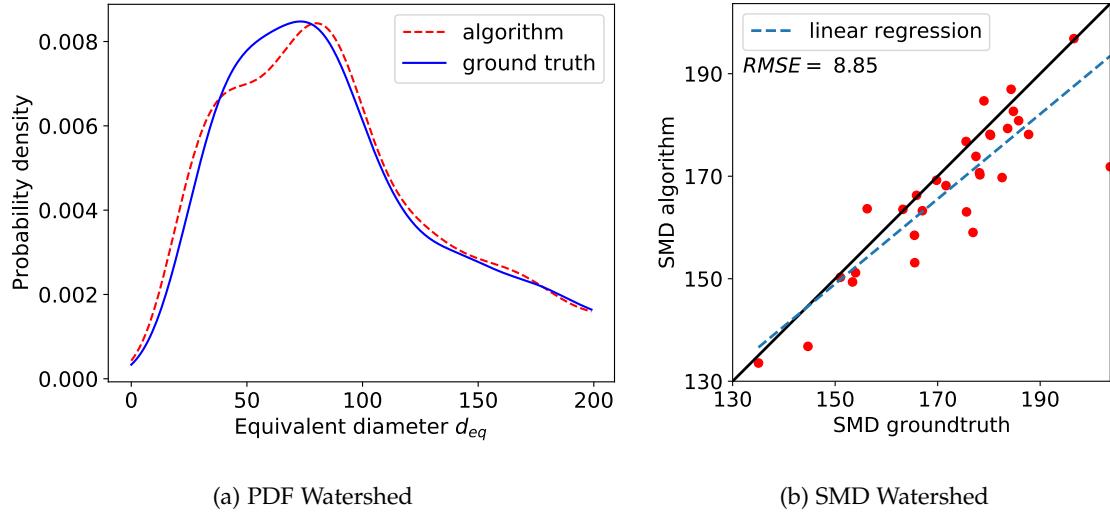


Figure 4.11.: Probability density function of the Watershed algorithm and real ground truth data (a). Scatter plot of the SMD values of the Hough algorithm and real ground truth data (b).

Precision (0.901) and recall (0.879) of the Watershed algorithm are both considerably higher than the results of the Hough algorithm. Looking at the example segmentation (see figure 4.12) a good match between bubble contours and the Watershed output (magenta ellipses) is noticeable. The probability density function (see figure 4.11a) resembles the ground truth distribution closer. Bubbles in the diameter range between 40 to 80 pixels are underestimated and bubbles between above 80 pixels slightly overestimated. The reason for this underestimation/overestimation is that overlapping bubble clusters with a circular shape are often wrongly detected as one big bubble. Additionally there is an over-segmentation of elongated bubbles, due to wrong marker selection using the euclidean distance, explaining the small overestimation of bubbles in the diameter range below 40 pixels. The area overlap (AJSC value) of 0.746 further validates the increasing performance of the Watershed algorithm when compared to the Hough algorithm. The regression line of the SMD scatter plot (see figure 4.9b) is a close match to the identity line, only having a slightly lower slope, due to missed bubbles. The RMSE of 8.85 confirms the good performance of the watershed algorithm. At last hypothesis testing, analogous to the description in 3.2.5, is conducted. The test statistic T , with a standard deviation of 0.037, results in a value of -2.05. Because $T < z = 1.64$ the hypothesis is not met and therefore the Watersehd algorithm not able to extract the BSD accurately.

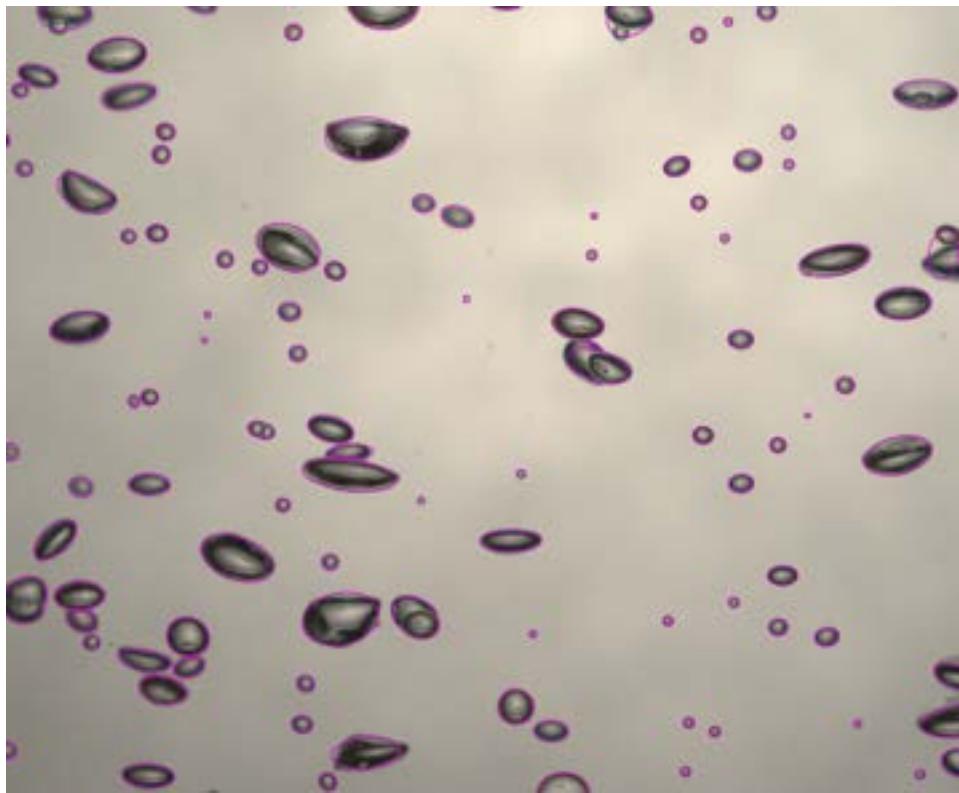


Figure 4.12.: Segmentation result of one of the camera images with the Watershed algorithm.
Detected ellipses are marked in magenta.

4.5.3. Concavepoint

This subsection compares the output of the Concavepoint algorithm on the real data set. Table 4.6 lists the mean of each performance measure.

Table 4.8.: Performance of the Concavepoint algorithm on the real data set.

algorithm	f1-score	precision	recall	AJSC
concavepoint	0.924	0.931	0.918	0.777

The implementation of the algorithm based on concavity outperforms the other two algorithms with a f1-score of 0.924, exceeding 0.9, and a AJSC value of 0.77. The root mean square has the lowest value ($RMSE = 7.9$) among the three algorithms. This also confirms the good performance of the Concavepoint algorithm. Furthermore the probability density function fits closely to the ground truth distribution (see figure 4.13a). However the Concavepoint distribution is shifted towards larger equivalent diameters indicating that not all bubble

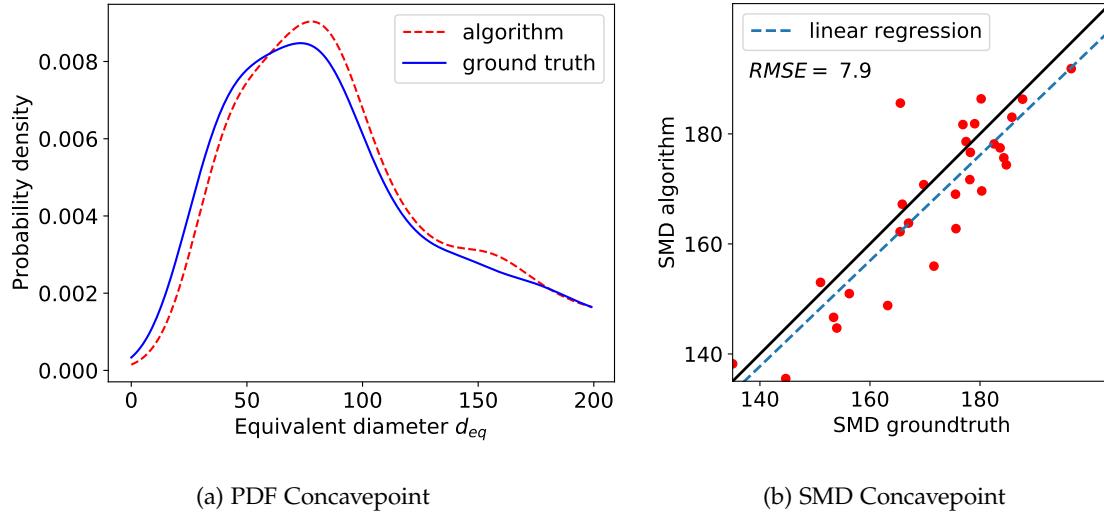


Figure 4.13.: Probability density function of the Concavepoint algorithm and real ground truth data (a). Scatter plot of the SMD values of the Hough algorithm and real ground truth data (b).

clusters are resolved correctly leading to less smaller bubbles and more bigger bubbles. At last hypothesis testing is done for the Concavepoint algorithm. With the f1-score of 0.924, a standard deviation of 0.029, and a sample size of $n = 30$ the test statistic T results in a value of 4.52. Because $T = 4.52 > z = 1.64$ the null hypothesis can be rejected and it can be concluded that the Concavepoint is able to extract the BSD well from the camera images.

4. Results and discussion

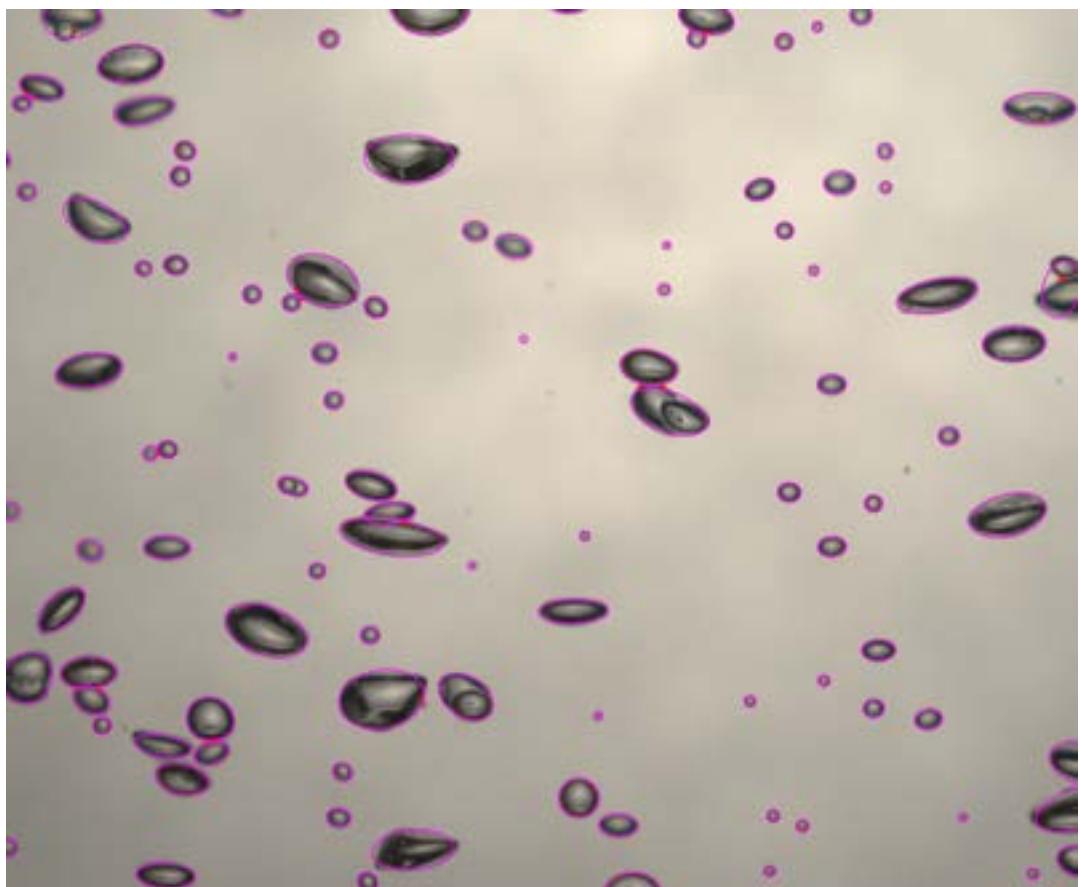


Figure 4.14.: Segmentation result of one of the camera images with the Concavepoint algorithm. Detected ellipses are marked in magenta.

5. Conclusion

In summary this thesis compared three state-of-art image processing methods in their segmentation performance and ability to extract two-dimensional bubble distribution. Hence for each method an algorithm was implemented based on recent literature. For evaluation purpose, the different algorithms were tested against both synthetic and real data. The results of the evaluation revealed that the Concavepoint algorithm achieved the highest performance and proved to meet the hypothesis in being able to depict the bubble size distribution for both data sets well. However the most suiting technique/algorithm depends heavily on the application and problem to be solved and can therefore not be universally determined.

One major constraint of all algorithms was an obligatory parameter selection necessary to achieve a high performance. This required heavy user interaction, therefore preventing automation, and made these methods specific for the present experimental environment and conditions. Different experimental environments and conditions and the applicability of the three methods against them were not studied.

In order to study the mass transfer, whose importance for liquid-gas chemical reactions has been asserted earlier, information of the interfacial area is key. In the scope of this thesis only the two dimensional projections of bubbles extracted from images were investigated. In what extent this enables the relation to the three dimensional parameters of bubbles (e.g. interfacial area) has not been quantified.

Regardless of the mentioned short-comings, the main goal of providing a comparison between prevalent image processing concepts in the specific application of extraction two-dimensional bubble distributions has been accomplished.

A. General Addenda

A.1. List of materials used

- 10 l plastic tank with the dimensions(l×w×h): 30 cm × 20 cm × 20 cm
- Plastic tube with a diameter of 9 mm and 37 holes
- Pump: Pontec PondoAir Set 450 $\dot{Q}_{max} = 450 \text{ l/h}$
- Camera: Panasonic DMC-GX8
- Light: Luceco Slimline Work Light 3000 lm

List of Figures

2.1. Illustration of a digital image f . There is one gray pixel with the intensity of 80 at the position $x_0 = 11, y_0 = 15$	5
2.2. Illustration of 4/8-connectivity. Connectivity is marked in green with respect to the center pixel in red.	6
2.3. Convolution of an image f with a kernel ω . The convolution is applied to the pixel in green. All pixel values underneath the blue kernel are multiplied with the respective kernel value. The sum of all products results in the new pixel value (marked in red).	7
2.4. The left column illustrates a binary image with an object (blue square). The right column shows the object after dilatation (a) and erosion (b). Figure adapted from [GW18].	8
2.5. Intensity histogram with a threshold value T marked in red. Figure adapted from [GW18].	9
3.1. Jaccard Similarity coefficient (JSC) calculation between a segmented object O_s and a ground truth object O_g . The area in blue represents the intersection and union respectively between the two objects.	13
3.2. Experimental set-up for the creation of the real data set.	15
3.3. Illustration of the labeling process for one of the camera images.	17
3.4. Example of a synthetically generated image.	18
3.5. Overview of the preprocessing procedure.	19
3.6. Functionality of the median filter. The 3×3 median kernel is applied to the pixel highlighted in green. The median value within the specified neighborhood is selected and replaces the initial value. The new pixel value after the application of the median value is marked in red.	20
3.7. Hole-filling process with and without prior dilatation.	21
3.8. Histogram of objects with a size between 0 and 100 pixels with and without the opening operation in advance.	22

3.9.	A circle with two points p_1, p_2 in an image space visualized in the transform space. The point of intersection between the two cones indicates the circle parameters. Figure adapted from [Kie92].	23
3.10.	A simple binary image of two overlapping circles with the corresponding distance map. The distance map is also visualized as a topographic map with the intensity as the z-axis.	25
3.11.	Euclidean distance with respect to the center of the square.	26
3.12.	Watershed with and without markers as seeding points. Heavy over-segmentation results without the selection of markers.	27
3.13.	Concavepoint determination process: The green lines are the detected contours from the binary image that are iterated to find concavepoints. Red points are the resulting four concavepoints.	28
4.1.	Probability density function of the equivalent diameter for the real data set. . .	30
4.2.	Some visual artifacts in the camera images that prove problematic for the manual marking process.	31
4.3.	Probability density function of the equivalent diameter for the synthetic data set. .	32
4.4.	Intensity distribution across the centerline (marked in black) of a real bubble (4.4a) and of a synthetic bubble (4.4b).	33
4.5.	Performance of the different parameters of the Hough algorithm. The best performing value is annotated for each parameter.	34
4.6.	Performance of the different parameters of the Watershed algorithm. The best performing value is annotated for each parameter.	36
4.7.	Performance of the different parameters of the Concavepoint algorithm. The best performing value is annotated for each parameter.	37
4.8.	Probability density function of the three algorithms and synthetic ground truth data. Scatter plot of the SMD values of the three algorithms and synthetic ground truth data.	40
4.9.	Probability density function of the Hough algorithm and real ground truth data (a). Scatter plot of the SMD values of the Hough algorithm and real ground truth data (b).	41
4.10.	Segmentation result of one of the camera images with the Hough algorithm. Detected circles are marked in magenta.	42
4.11.	Probability density function of the Watershed algorithm and real ground truth data (a). Scatter plot of the SMD values of the Hough algorithm and real ground truth data (b).	43

List of Figures

4.12. Segmentation result of one of the camera images with the Watershed algorithm. Detected ellipses are marked in magenta.	44
4.13. Probability density function of the Concavepoint algorithm and real ground truth data (a). Scatter plot of the SMD values of the Hough algorithm and real ground truth data (b).	45
4.14. Segmentation result of one of the camera images with the Concavepoint algo- rithm. Detected ellipses are marked in magenta.	46

List of Tables

4.1.	Parameters for the Hough algorithm.	35
4.2.	Parameters for the Watershed algorithm.	35
4.3.	Parameters for the Concavepoint algorithm.	38
4.4.	Performance of the the algorithms on the synthetic data set.	38
4.5.	Test statistic T for all algorithms of the synthetic data set.	39
4.6.	Performance of the Hough algorithm on the real data set.	39
4.7.	Performance of the Watershed algorithm on the real data set.	42
4.8.	Performance of the Concavepoint algorithm on the real data set.	44

Bibliography

- [LDK13] Y. M. Lau, N. G. Deen, and J. A. Kuipers. "Development of an image measurement technique for size distribution in dense bubbly flows". In: *Chemical Engineering Science* 94 (May 2013), pp. 20–29. ISSN: 00092509. doi: 10.1016/j.ces.2013.02.043.
- [Laa+05] M. Laakkonen, P. Moilanen, T. Miettinen, K. Saari, M. Honkanen, P. Saarenrinne, and J. Aittamaa. "Local bubble size distributions in agitated vessel comparison of three experimental techniques". In: *Chemical Engineering Research and Design* 83.1 A (2005), pp. 50–58. ISSN: 02638762. doi: 10.1205/cherd.04122.
- [Fer+12] A. Ferreira, G. Pereira, J. A. Teixeira, and F. Rocha. "Statistical tool combined with image analysis to characterize hydrodynamics and mass transfer in a bubble column". In: *Chemical Engineering Journal* 180 (Jan. 2012), pp. 216–228. ISSN: 13858947. doi: 10.1016/j.cej.2011.09.117.
- [Kul07] A. A. Kulkarni. "Mass transfer in bubble column reactors: Effect of bubble size distribution". In: *Industrial and Engineering Chemistry Research* 46.7 (Mar. 2007), pp. 2205–2211. ISSN: 08885885. doi: 10.1021/ie061015u. URL: <https://pubs.acs.org/doi/abs/10.1021/ie061015u>.
- [CM61] P. Calderbank and M. Moo-Young. "The continuous phase heat and mass-transfer properties of dispersions". In: *Chemical Engineering Science* 16.1 (1961), pp. 39–54. ISSN: 0009-2509. doi: [https://doi.org/10.1016/0009-2509\(61\)87005-X](https://doi.org/10.1016/0009-2509(61)87005-X). URL: <http://www.sciencedirect.com/science/article/pii/000925096187005X>.
- [Zho+16] S. Zhong, X. Zou, Z. Zhang, and H. Tian. "A flexible image analysis method for measuring bubble parameters". In: *Chemical Engineering Science* 141 (Feb. 2016), pp. 143–153. ISSN: 00092509. doi: 10.1016/j.ces.2015.10.033.
- [Kum+76] A. Kumar, T. E. Degaleesan, G. S. Laddha, and H. E. Hoelscher. "Bubble swarm characteristics in bubble columns". In: *The Canadian Journal of Chemical Engineering* 54.5 (1976), pp. 503–508. ISSN: 1939019X. doi: 10.1002/cjce.5450540525.

Bibliography

- [Yia+88] J. B. Yianatos, J. A. Finch, G. S. Dobby, and M. Xu. "Bubble size estimation in a bubble swarm". In: *Journal of Colloid And Interface Science* 126.1 (1988), pp. 37–44. ISSN: 00219797. DOI: 10.1016/0021-9797(88)90096-3.
- [Maa+11] S. Maafß, S. Wollny, A. Voigt, and M. Kraume. "Experimental comparison of measurement techniques for drop size distributions in liquid/liquid dispersions". In: *Experiments in Fluids* 50.2 (2011), pp. 259–269.
- [BG91] M. Barigou and M. Greaves. "A capillary suction probe for bubble size measurement". In: *Measurement Science and Technology* 2.4 (1991), pp. 318–326. ISSN: 09570233. DOI: 10.1088/0957-0233/2/4/009.
- [Wan+18] W. Wang, X. Yuan, X. Wu, T. Ji, and L. Feng. "Separating Touching Particles: A Concavity-Based Method Using the Area Ratio of a Circular Mask". In: *IEEE Systems, Man, and Cybernetics Magazine* 4.2 (2018), pp. 24–32. ISSN: 2380-1298. DOI: 10.1109/msmc.2018.2794559.
- [ELR05] M. K. El-Kamash, M. R. Loewen, and N. Rajaratnam. "Measurements of void fraction and bubble properties on a stepped chute using a fiber-optic probe". In: *Canadian Journal of Civil Engineering* 32.4 (2005), pp. 636–643. ISSN: 03151468. DOI: 10.1139/l05-014.
- [RL07] G. Rojas and M. R. Loewen. "Fiber-optic probe measurements of void fraction and bubble size distributions beneath breaking waves". In: *Experiments in Fluids* 43.6 (2007), pp. 895–906. ISSN: 07234864. DOI: 10.1007/s00348-007-0356-5.
- [Pra08] H. M. Prasser. "Novel experimental measuring techniques required to provide data for CFD validation". In: *Nuclear Engineering and Design* 238.3 (2008), pp. 744–770. ISSN: 00295493. DOI: 10.1016/j.nucengdes.2007.02.050.
- [Bus+13] A. Busciglio, F. Grisafi, F. Scargiali, and A. Brucato. "On the measurement of local gas hold-up, interfacial area and bubble size distribution in gas-liquid contactors via light sheet and image analysis: Imaging technique and experimental results". In: *Chemical Engineering Science* 102 (Oct. 2013), pp. 551–566. ISSN: 00092509. DOI: 10.1016/j.ces.2013.08.029.
- [FL16] Y. Fu and Y. Liu. "Development of a robust image processing technique for bubbly flow measurement in a narrow rectangular channel". In: *International Journal of Multiphase Flow* 84 (2016), pp. 217–228. ISSN: 03019322. DOI: 10.1016/j.ijmultiphaseflow.2016.04.011. URL: <http://dx.doi.org/10.1016/j.ijmultiphaseflow.2016.04.011>.

Bibliography

- [BDW02] C. Boyer, A. M. Duquenne, and G. Wild. "Measuring techniques in gas-liquid and gas-liquid-solid reactors". In: *Chemical Engineering Science* 57.16 (2002), pp. 3185–3215. ISSN: 00092509. doi: 10.1016/S0009-2509(02)00193-8.
- [SF86] T. Y. Sun and G. M. Faeth. "Structure of turbulent bubbly jets-I. Methods and centerline properties". In: *International Journal of Multiphase Flow* 12.1 (1986), pp. 99–114. ISSN: 03019322. doi: 10.1016/0301-9322(86)90006-6.
- [Mar+81] W. W. Martin, A. H. Adbelmessih, J. J. Liska, and F. Durst. "Characteristics of laser-doppler signals from bubbles". In: *International Journal of Multiphase Flow* 7.4 (1981), pp. 439–460. ISSN: 03019322. doi: 10.1016/0301-9322(81)90049-5.
- [Tso+01] N. A. Tsochatzidis, P. Guiraud, A. M. Wilhelm, and H. Delmas. "Determination of velocity, size and concentration of ultrasonic cavitation bubbles by the phase-Doppler technique". In: *Chemical Engineering Science* 56.5 (2001), pp. 1831–1840. ISSN: 00092509. doi: 10.1016/S0009-2509(00)00460-7.
- [SL18] K. Song and Y. Liu. "A compact x-ray system for two-phase flow measurement". In: *Measurement Science and Technology* 29.2 (2018). ISSN: 13616501. doi: 10.1088/1361-6501/aaa283.
- [Bie+13] A. Bieberle, H. U. Härtling, S. Rabha, M. Schubert, and U. Hampel. "Gamma-ray computed tomography for imaging of multiphase flows". In: *Chemie-Ingenieur-Technik* 85.7 (2013), pp. 1002–1011. ISSN: 0009286X. doi: 10.1002/cite.201200250.
- [Str+16] N. Strokina, J. Matas, T. Eerola, L. Lensu, and H. Kälviäinen. "Detection of bubbles as concentric circular arrangements". In: *Machine Vision and Applications* 27.3 (Apr. 2016), pp. 387–396. ISSN: 14321769. doi: 10.1007/s00138-016-0749-7. URL: <https://link.springer.com/article/10.1007/s00138-016-0749-7>.
- [Bet+20] A. Bettaieb, N. Filali, T. Filali, and H. B. Aissia. "Image processing, pattern recognition an efficient algorithm for overlapping bubbles segmentation". In: *Computer Optics* 44.3 (2020), pp. 363–374. ISSN: 24126179. doi: 10.18287/2412-6179-C0-605.
- [Lan+18] M. de Langlard, H. Al-Saddik, S. Charton, J. Debayle, and F. Lamadie. "An efficiency improved recognition algorithm for highly overlapping ellipses: Application to dense bubbly flows". In: *Pattern Recognition Letters* 101 (2018), pp. 88–95. ISSN: 01678655. doi: 10.1016/j.patrec.2017.11.024. URL: <https://doi.org/10.1016/j.patrec.2017.11.024>.

Bibliography

- [Tab+06] B. Taboada, L. Vega-Alvarado, M. S. Córdova-Aguilar, E. Galindo, and G. Corkidi. "Semi-automatic image analysis methodology for the segmentation of bubbles and drops in complex dispersions occurring in bioreactors". In: *Experiments in Fluids* 41.3 (2006), pp. 383–392. ISSN: 07234864. DOI: 10.1007/s00348-006-0159-0.
- [Brá+09] L. M. R. Brás, E. F. Gomes, M. M. M. Ribeiro, and M. M. L. Guimarães. "Drop Distribution Determination in a Liquid-Liquid Dispersion by Image Processing". In: *International Journal of Chemical Engineering* 746439 (2009). DOI: 10.1155/2009/746439.
- [Mey94] F. Meyer. *SIGNAL PROCESSING Topographic distance and watershed lines*. Tech. rep. 1994, pp. 113–125.
- [BL00] A. Bleau and L. J. Leon. "Watershed-based segmentation and region merging". In: *Computer Vision and Image Understanding* 77.3 (2000), pp. 317–370. ISSN: 10773142. DOI: 10.1006/cviu.1999.0822.
- [Kar+15] A. Karn, C. Ellis, R. Arndt, and J. Hong. "An integrative image measurement technique for dense bubbly flows with a wide size distribution". In: *Chemical Engineering Science* 122 (Jan. 2015), pp. 240–249. ISSN: 00092509. DOI: 10.1016/j.ces.2014.09.036.
- [Hon+05] M. Honkanen, P. Saarenrinne, T. Stoor, and J. Niinimäki. "Recognition of highly overlapping ellipse-like bubble images". In: *Measurement Science and Technology* 16.9 (July 2005), p. 1760. ISSN: 0957-0233. DOI: 10.1088/0957-0233/16/9/007.
- [ZJL12] W. H. Zhang, X. Jiang, and Y. M. Liu. "A method for recognizing overlapping elliptical bubbles in bubble image". In: *Pattern Recognition Letters* 33.12 (Sept. 2012), pp. 1543–1548. ISSN: 01678655. DOI: 10.1016/j.patrec.2012.03.027.
- [Zaf+15a] S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, and H. Haario. "Segmentation of Overlapping Elliptical Objects in Silhouette Images". In: *IEEE Transactions on Image Processing* 24.12 (Dec. 2015), pp. 5942–5952. ISSN: 10577149. DOI: 10.1109/TIP.2015.2492828.
- [Far+12] M. Farhan, O. Yli-Harja, A. Niemistö, and N. Niemistö. "A novel method for splitting clumps of convex objects incorporating image intensity and using rectangular window-based concavity point-pair search". In: *Pattern Recognition* 46 (2012), pp. 741–751. DOI: 10.1016/j.patcog.2012.09.008. URL: <http://dx.doi.org/10.1016/j.patcog.2012.09.008>.

Bibliography

- [Gra97] I. Grant. "Particle image velocimetry: A review". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 211.1 (1997), pp. 55–76. ISSN: 09544062. DOI: 10.1243/0954406971521665.
- [Cer+18] R. F. Cerqueira, E. E. Paladino, B. K. Ynumaru, and C. R. Maliska. "Image processing techniques for the measurement of two-phase bubbly pipe flows using particle image and tracking velocimetry (PIV/PTV)". In: *Chemical Engineering Science* 189 (2018), pp. 1–23. ISSN: 00092509. DOI: 10.1016/j.ces.2018.05.029. URL: <https://doi.org/10.1016/j.ces.2018.05.029>.
- [Mar+10] J. Martnez Mercado, D. Chehata Gmez, D. Van Gils, C. Sun, and D. Lohse. "On bubble clustering and energy spectra in pseudo-turbulence". In: *Journal of Fluid Mechanics* 650.1991 (2010), pp. 287–306. ISSN: 00221120. DOI: 10.1017/S0022112009993570. arXiv: 0912.1620.
- [GW18] R. C. Gonzalez and R. E. Woods. *Digital Image Processing (4th Ed)*. 2018, p. 1022. ISBN: 978-1-292-22304-9.
- [Joh16] F. B. N. John C. Russ. *The Image Processing Handbook*. 2016. ISBN: 9781498740289.
- [RW03] A. W. R. Fisher, S. Perkins and E. Wolfart. *Morphology*. 2003. URL: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm> (visited on 11/18/2020).
- [Bra00] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).
- [Ped+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. URL: <http://jmlr.org/papers/v12/pedregosa11a.html>.
- [Van+14] S. Van Der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. "Scikit-image: Image processing in python". In: *PeerJ* 2014.1 (June 2014), e453. ISSN: 21678359. DOI: 10.7717/peerj.453. URL: <https://developers.google.com/>.
- [Gil+–] S. Gillies et al. *Shapely: manipulation and analysis of geometric objects*. toblerity.org, 2007–. URL: <https://github.com/Toblerity/Shapely>.
- [Vir+20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold,

- R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. doi: 10.1038/s41592-019-0686-2.
- [Har+20] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. “Array programming with NumPy”. In: *Nature* 585 (2020), pp. 357–362. doi: 10.1038/s41586-020-2649-2.
- [McK10] W. McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by S. van der Walt and J. Millman. 2010, pp. 56–61. doi: 10.25080/Majora-92bf1922-00a.
- [Hun07] J. D. Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science Engineering* 9.3 (2007), pp. 90–95. doi: 10.1109/MCSE.2007.55.
- [Men+05] P. C. Mena, M. N. Pons, J. A. Teixeira, and F. A. Rocha. “Using image analysis in the study of multiphase gas absorption”. In: *Chemical Engineering Science* 60.18 (Sept. 2005), pp. 5144–5150. issn: 00092509. doi: 10.1016/j.ces.2005.04.049.
- [Sco15] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- [PMN98] A. W. Pacek, C. C. Man, and A. W. Nienow. “On the Sauter mean diameter and size distributions in turbulent liquid/liquid dispersions in a stirred vessel”. In: *Chemical Engineering Science*. Vol. 53. 11. Elsevier Sci Ltd, June 1998, pp. 2005–2011. doi: 10.1016/S0009-2509(98)00068-2.
- [KD16] P. B. Kowalcuk and J. Drzymala. “Physicochemical Problems of Mineral Processing View project Copper-bearing shale View project Physical meaning of the Sauter mean diameter of spherical particulate matter”. In: 34.6 (2016), pp. 645–647. doi: 10.1080/02726351.2015.1099582. url: <http://dx.doi.org/10.1080/02726351.2015.1099582>.
- [Jun06] B. Junker. “Measurement of bubble and pellet size distributions: Past and current image analysis technology”. In: *Bioprocess and Biosystems Engineering* 29.3 (2006), pp. 185–206. issn: 16157591. doi: 10.1007/s00449-006-0070-3.

Bibliography

- [Hol17] S. Holmes. *RMS Error*. 2017. doi: 10.1007/978-3-319-17885-1_101118. URL: <https://statweb.stanford.edu/%7B~%7Dsusan/courses/s60/split/node60.html> (visited on 11/30/2020).
- [CCT09] S. S. Choi, S. H. Cha, and C. C. Tappert. *A survey of binary similarity and distance measures*. Tech. rep. 2009, pp. 80–85.
- [Zaf+15b] S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, and H. Haario. *Segmentation of partially overlapping nanoparticles using concave points*. Tech. rep. 2015, pp. 187–197. doi: 10.1007/978-3-319-27857-5_17.
- [Fat+10a] H. Fatakdawala, J. Xu, A. Basavanhally, G. Bhanot, S. Ganesan, M. Feldman, J. E. Tomaszewski, and A. Madabhushi. “Expectation-maximization-driven geodesic active contour with overlap resolution (EMaGACOR): Application to lymphocyte segmentation on breast cancer histopathology”. In: *IEEE Transactions on Biomedical Engineering* 57.7 (2010), pp. 1676–1689. issn: 00189294. doi: 10.1109/TBME.2010.2041232.
- [Ilo+18] J. Ilonen, R. Juránek, T. Eerola, L. Lensu, M. Dubská, P. Zemčík, and H. Kälviäinen. “Comparison of bubble detectors and size distribution estimators”. In: *Pattern Recognition Letters* 101 (Jan. 2018), pp. 60–66. issn: 01678655. doi: 10.1016/j.patrec.2017.11.014.
- [OR18] M. Oestreich and O. Romberg. “Ende gut, alles gut!” In: *Keine Panik vor Statistik!* Springer, 2018, pp. 277–277.
- [FPF99] A. Fitzgibbon, M. Pilu, and R. B. Fisher. “Direct least square fitting of ellipses”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5 (1999), pp. 476–480. issn: 01628828. doi: 10.1109/34.765658.
- [MC04] K. F. Mulchrone and K. R. Choudhury. “Fitting an ellipse to an arbitrary shape: Implications for strain analysis”. In: *Journal of Structural Geology* 26.1 (2004), pp. 143–153. issn: 01918141. doi: 10.1016/S0191-8141(03)00093-2.
- [Lau+13] Y. M. Lau, K. T. Sujatha, M. Gaeini, N. G. Deen, and J. A. Kuipers. “Experimental study of the bubble size distribution in a pseudo-2D bubble column”. In: *Chemical Engineering Science* 98 (July 2013), pp. 203–211. issn: 00092509. doi: 10.1016/j.ces.2013.05.024.
- [Vil+19] L. R. Villegas, D. Colombet, P. Guiraud, D. Legendre, S. Cazin, and A. Cockx. “Image processing for the experimental investigation of dense dispersed flows: application to bubbly flows”. In: *International Journal of Multiphase Flow* 111

Bibliography

- (2019), pp. 16–30. doi: 10.1016/j.ijmultiphaseflow.2018.10.017. URL: <https://doi.org/10.1016/j.ijmultiphaseflow.2018.10.017>.
- [ASS] K. Abhinav, J. Singh Chauhan, and D. Sarkar. *Image Segmentation of Multi-Shaped Overlapping Objects*. Tech. rep. arXiv: 1711.02217v1.
- [ZH12] Y. Zhu and C. Huang. “2012 International Conference on Solid State Devices and Materials Science An Improved Median Filtering Algorithm for Image Noise Reduction * peer-review under responsibility of [name organizer]”. In: *Physics Procedia* 25 (2012), pp. 609–616. doi: 10.1016/j.phpro.2012.03.133. URL: www.sciencedirect.com.
- [Jai89] A. K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [Hou62] P. V. Hough. *Method and means for recognizing complex patterns*. US Patent 3,069,654. Dec. 1962.
- [DH72] R. O. Duda and P. E. Hart. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Tech. rep. 1. 1972, pp. 11–15. doi: 10.1145/361237.361242.
- [Dav87] E. R. Davies. “A high speed algorithm for circular object location”. In: *Pattern Recognition Letters* 6.5 (Dec. 1987), pp. 323–333. issn: 01678655. doi: 10.1016/0167-8655(87)90015-8.
- [Har09] P. E. Hart. “How the Hough transform was invented [DSP History]”. In: *IEEE Signal Processing Magazine* 26.6 (2009), pp. 18–22. issn: 10535888. doi: 10.1109/MSP.2009.934181.
- [Bal81] D. H. Ballard. *GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES**. Tech. rep. 2. 1981, pp. 20111–20123.
- [MC15] P. Mukhopadhyay and B. B. Chaudhuri. “A survey of Hough Transform”. In: *Pattern Recognition* 48.3 (Mar. 2015), pp. 993–1010. issn: 00313203. doi: 10.1016/j.patcog.2014.08.027.
- [XOK90] L. Xu, E. Oja, and P. Kultanen. “A new curve detection method: randomized Hough transform (RHT)”. In: *Pattern recognition letters* 11.5 (1990), pp. 331–338.
- [XO09] L. Xu and E. Oja. “Randomized hough transform”. In: *Encyclopedia of Artificial Intelligence*. IGI Global, 2009, pp. 1343–1350.
- [KBS75] C. Kimme, D. Ballard, and J. Sklansky. *Finding Circles by an Array of Accumulators*. Tech. rep. 1975.

- [Kie92] P. Kierkegaard. "A method for detection of circular arcs based on the Hough transform". In: *Machine Vision and Applications* 5.4 (1992), pp. 249–263. ISSN: 09328092. doi: 10.1007/BF01212714.
- [MG98] A. N. Moga and M. Gabbouj. "Parallel Marker-Based Image Segmentation with Watershed Transformation". In: *Journal of Parallel and Distributed Computing* 51.1 (1998), pp. 27–45. ISSN: 07437315. doi: 10.1006/jpdc.1998.1448.
- [Beu79] S. Beucher. "Use of watersheds in contour detection". In: *Proceedings of the International Workshop on Image Processing*. CCETT. 1979.
- [MB90] F. Meyer and S. Beucher. "Morphological segmentation". In: *Journal of Visual Communication and Image Representation* 1.1 (1990), pp. 21–46. ISSN: 10959076. doi: 10.1016/1047-3203(90)90014-M.
- [Att+16] T. Atta-Fosu, W. Guo, D. Jeter, C. Mizutani, N. Stopczynski, and R. Sousa-Neves. "3D Clumped Cell Segmentation Using Curvature Based Seeded Watershed". In: *Journal of Imaging* 2.4 (Nov. 2016), p. 31. ISSN: 2313-433X. doi: 10.3390/jimaging2040031. URL: <http://www.mdpi.com/2313-433X/2/4/31>.
- [RM01] J. B. T. M. Roerdink and A. Meijster. *The Watershed Transform: Definitions, Algorithms and Parallelization Strategies*. Tech. rep. 2001, pp. 187–228.
- [GB09] M. A. Gonzalez and V. L. Ballarin. "Automatic marker determination algorithm for watershed segmentation using clustering". In: *Latin American Applied Research* 39.3 (2009), pp. 225–229. ISSN: 03270793.
- [CR09] J. Cheng and J. C. Rajapakse. "Segmentation of clustered nuclei with shape markers and marking function". In: *IEEE Transactions on Biomedical Engineering* 56.3 (2009), pp. 741–748. ISSN: 00189294. doi: 10.1109/TBME.2008.2008635.
- [Beu94] S. Beucher. "Watershed, hierarchical segmentation and waterfall algorithm". In: *Mathematical morphology and its applications to image processing*. Springer, 1994, pp. 69–76.
- [Per+17] B. Perret, J. Cousty, S. J. F. Guimaraes, and D. S. Maia. "Evaluation of hierarchical watersheds". In: *IEEE Transactions on Image Processing* 27.4 (2017), pp. 1676–1688.
- [CYP04] Q. Chen, X. Yang, and E. M. Petriu. *Watershed segmentation for binary images with different distance transforms*. Tech. rep. 2004, pp. 111–116. doi: 10.1109/have.2004.1391891.

Bibliography

- [MD02] K. N. Mohana Rao and A. G. Dempster. "Modification on distance transform to avoid over-segmentation and under-segmentation". In: *Proceedings VIPromCom 2002 - 4th EURASIP - IEEE Region 8 International Symposium on Video / Image Processing and Multimedia Communications* June (2002), pp. 295–301. doi: 10.1109/VIPROM.2002.1026672.
- [Fat+10b] H. Fatakdawala, J. Xu, A. Basavanhally, G. Bhanot, S. Ganesan, M. Feldman, J. E. Tomaszewski, and A. Madabhushi. "Expectation-maximization-driven geodesic active contour with overlap resolution (EMaGACOR): Application to lymphocyte segmentation on breast cancer histopathology". In: *IEEE Transactions on Biomedical Engineering* 57.7 (2010), pp. 1676–1689. issn: 00189294. doi: 10.1109/TBME.2010.2041232.
- [Suz85] S. Suzuki. *Topological Structural Analysis of Digitized Binary Images by Border Following*. Tech. rep. 1985, pp. 32–46.