KIEL UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE

Master's Thesis

Underwater Bubble Shape Measurement and Analysis

Claudius Zelenka

January 21, 2014

Supervised by:
Prof. Dr.-Ing. Reinhard Koch
Dipl.-Inf. Anne Jordt

Abstract

Der Einfluss von Unterwasser-Gasautritten auf globale Klima- und Ökosysteme ist ein wichtiges Thema in der Klimatologie und Meeresforschung. Die austretenden Treibhausgase, meistens Kohlendioxid und Methan, haben einen grundlegenden Einfluss auf die Biosphäre, daher sind genaue Messmethoden erforderlich.

In dieser Arbeit werden verschiedene Methoden zur Detektion und Vermessung der Form und Bewegung von Gasblasen entwickelt. Die Detektion von Gasblasen erfolgt mit einem erweiterten Snakeverfahren und dem Fitting einer Ellipse mittels des CMA-ES Algorithmus. Verschiedene Fitnessfunktionen für das CMA-ES Verfahren und eine optimale Wahl der Parameter für das Snakeverfahren wurden untersucht. Für die Verfolgung der Bewegung der Gasblasen werden zweierlei Ansätze genutzt: Zum einen die Ermittlung der durchschnittlichen Steiggeschwindigkeit, zum anderen eine individuelle Verfolgung der Gasblasen mit Kalman-Prädiktion und einer anschließenden Ermittlung der Blasenbewegung durch einen Matching-Algorithmus. Die verschiedenen Verfahren werden in einem Versuchsaufbau getestet und analysiert, sowie mit Ground-Truth Daten verglichen.

Aus den Ergebnissen der Algorithmen werden Statistiken über die Verteilung der Größe und Geschwindigkeit der Blasen zusammengestellt. Anschließend werden die Gasblasen mit Ellipsoiden approximiert und das Volumen des Blasenstroms in Liter pro Minute berechnet. Der Hauptbeitrag dieser Arbeit besteht in dem Vergleich und der Verbesserung von Snake und CMA-ES basierenden Algorithmen zur Detektion von Gasblasen.

Abstract

The influence of water gas seeps on the global climate and ecosystems is an important topic in climatology and marine research. The emitted greenhouse gases, mostly carbon dioxide and methane, have essential impact on the biosphere. Therefore exact measurement methods are required.

In this thesis different methods for the detection and measurement of bubble shape and motion from bubble flow images are developed. For the bubble shape detection enhanced snake methods and CMA-ES optimization based ellipse fitting methods are employed. Various fitness functions for the CMA-ES method and the optimal choice of parameters for the snake algorithm were studied. To track the bubble motion two different approaches are employed: Firstly a mean velocity approach measuring the mean rise velocity and secondly individual bubble tracking with Kalman prediction and determination of the bubble motion using a best matching algorithm.
The different detection methods are tested and analyzed in an experimental setup, as well as compared against ground truth data. Statistics about the distribution of bubble shapes and velocities in a bubble stream are evaluated and the total flux volume in liter per minute is measured. The main contribution of this work is the comparison and improvement of snake and CMA-ES fitting algorithms for bubble detection.

Declaration

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.
Weiterhin versichere ich, dass diese Arbeit noch nicht als Abschlussarbeit an anderer Stelle vorgelegen hat.

Claudius Zelenka

Acknowledgments

I wish to thank Dipl.-Inf. Anne Jordt for the best possible supervision and many helpful suggestions. I am grateful to Prof. Dr.-Ing. Reinhard Koch for finding an exciting topic for my thesis.

Contents

List of Figures

1. Introduction

Underwater gas seeps play an important role in climatology and marine research. Modeling and understanding their complex impacts on the biosphere requires the knowledge of the composition and amount of the emitted gases and its absorption in the surrounding water.

Observing and analyzing the bubbles can help answering these questions. The main focus in recent research is on methane and carbon dioxide gases, as both are greenhouse gases and have thus the largest impact on climate change. Dissolved gas can essentially affect the chemical properties of the surrounding water (see [MBB+11]). Large amounts of solved carbon dioxide acidify the ocean, which has a significant impact on marine life as shown by [IKH+04]. Dissolved methane can also influence the marine life, particularly the fauna [DH92]. In recent years, much research into these consequences has been made. A better understanding of the impact on the environment requires exact knowledge about marine gas seeps and their interactions with the surrounding water.

If seeped gas is not immediately dissolved by the surrounding water, it forms bubbles which rise towards the surface and can be observed. The shape and velocity of the bubbles is dependent on the volume of the seeping gas, its properties and the properties of the surroundings. To observe the bubbles, this thesis uses image sequences from a camera. From these images the shape, motion and volume of the bubbles are obtained with image processing methods.

The goal of this thesis is to formulate suitable methods of image processing, which can be employed to extract the shape and motion of bubbles from image sequences. In context with this task, an experimental setup of a bubble measuring system (bubble meter), which delivers good images for further processing, is designed and built. In order to obtain information about bubble shape, the statistical evalution of the results is helpful.

The focus of this thesis will be on the detection of bubbles in camera images using appropriate image processing methods. Especially methods and algorithms based on

the snake algorithm, a general contour localization technique and ellipse fitting with the CMA-ES non-linear optimization algorithm will be developed and evaluated. Methods to track the bubbles and to measure their size and velocity over an image series will also be developed, to allow a better data gathering on marine gas seeps for future climatology and marine research. For verification of the accuracy and overall performance of the detection and tracking methods in combination the flux volume will be measured.

To summarize, the general motivation of this thesis is to develop and analyze different methods towards a better bubble meter. Bubble meters have growing areas of applications in science and industry. Some of them are listed below.

Marine Gas Seeps Marine gas seeps emit a number of different kinds of gases [MGA+06].Among them are carbon dioxide and methane, primary greenhouse gases. The influence of these gases on marine life and the climate [LCD98]. is an active research topic. Close to the methane seeps an increase in biomass and changes in fauna has been observed [DH92]. Bubble meters can help in locating, analyzing and quantifying these gas seeps as seen in [GAE+06]. Known areas featuring large gas seeps are for example the Panarea area near Sicily [MBB+11], the Tommeliten in the North Sea [SvDRG+11] and the Krim in the Caspian Sea [GAE+06].

Breaking Waves The surface of an ocean is an interface for gas exchange be-tween the ocean and the atmosphere. The bubbles created by breaking waves play a large part in the gas exchange. Their influence on carbon dioxide transfer is described in [LdLKHC03]. A bubble meter can help to understand better the gas exchange physics. Quantifying these interactions is important for the parameterization of climate models. [FW07].

Chemistry An application of a bubble meter in chemistry is the examination of vapor bubbles in seething water (see [CB03]). A bubble meter can also be used to examine the heat transfer between surfaces and liquids as in [CB06] and in many chemical reactions, which produce gaseous substances.

Food Industry As shown in [KYT+97] and [HHV04], the measurement of bubbles can help for example in the brewing process of beverages and champagne.

Flux Meter Bubbles in a fluid are a special case of two phase flow, in this case liquid and gaseous phase. Therefore some flow meters for multi-phase flow, can be

used as bubble meters, for example the flow meter described in [HHV04].

Medicine Air embolism (air bubbles in blood) can be a live threatening condition. [OAYT06] deals with discriminating dangerous larger bubbles from harmless micro bubbles. In [Bel80] the air bubbles in the blood flow formed by decompression are quantified.

Nuclear Reactor Safety As claimed by [Has01] the bubble meter developed in that research report can improve models and codes used in nuclear reactor safety.

1.1. Related Work

Gas bubbles have been a research subject for over 70 years as [Bry33] illustrates. Because of the multitude of applications and the long history, a large number of works has been published in this field.

As explained in the introduction, the main application for the research of this thesis is the observation of bubbles from marine gas seeps. This gas bubbles are often of unknown composition in an underwater setting. A selection over some of the published works is shown in Table 1.1.

According to published works the characteristics of bubbles are strongly dependent on their environment and their composition. A model trying to incorporate many factors is presented in [MGA$^+$06]. It includes many parameters, like the gas composition and diffusion coefficients for these gases, which are typically the results of this research and hence not known beforehand or while the gas bubbles are measured.

Figure 1.1 shows a diagram of the bubble size and corresponding uprising velocities composed by [GAE$^+$06], which were measured by various researchers referenced in [GAE$^+$06]. First of all it shows the wide range of observed bubble sizes . Very large bubbles occur seldom and are unstable (they decay into smaller bubbles) [LdLKHC03] and small bubbles ($< 2\text{mm}$) have no significant impact on the total flux volume, because the volume increases with the third exponent of the bubble radius. Hence the bubble size of interest in this thesis is set to be between 3 and 10 mm. Figure 1.1 shows that very large and very small bubbles adhere to Stokes law (eq. A.1) and the Hardamard–Rybczynski equation (eq. A.2), in agreement with the references from the table above [LK05] and [WD09a]. Therefore, for such bubbles, the speed can be determined accurately from its size. However, in the bubble size range of interest the bubble speed depends strongly on the environment and gas

Characteristics of Underwater Bubbles and Description of Conducted Research	Measurement Location	Reference
Short time bubble motion is chaotic.	Laboratory	[KYT+97]
Bubble size and gas diffusion to water differ with gas composition and depth	Near the shore of the Krim	[MGA+06]
Carbon dioxide bubble sizes in shallow water	Panaera area near Sicily	[MBB+11]
Bubble size differs for different media. Salt water and fresh water are compared.	Laboratory	[PJK04]
Bubble size distribution is influenced by various chemicals	Laboratory	[LLZR11]
Bubble size and rising velocity is strongly dependent on alcohol, organic acid and potassium chloride concentration.	Laboratory	[JMS92]
Oily bubbles show changed properties, much lower rising velocity.	California	[LB05]
Quantification of methane gas emissions, bubble diameter and methane concentration in bubbles in different depths.	Tommelitten in the north sea	[SvDRG+11]
For micro bubbles the relationship between bubble rise velocity and size can be explained with Stokes law (see A.1), and the Hadamard equation (see A.2).	Laboratory with X-ray illumination	[LK05]
Bubble size and uprising velocity of single air bubbles comply with Stokes law and the Hadamard equation.	Laboratory	[WD09a]

Table 1.1.: Overview over selected related works.

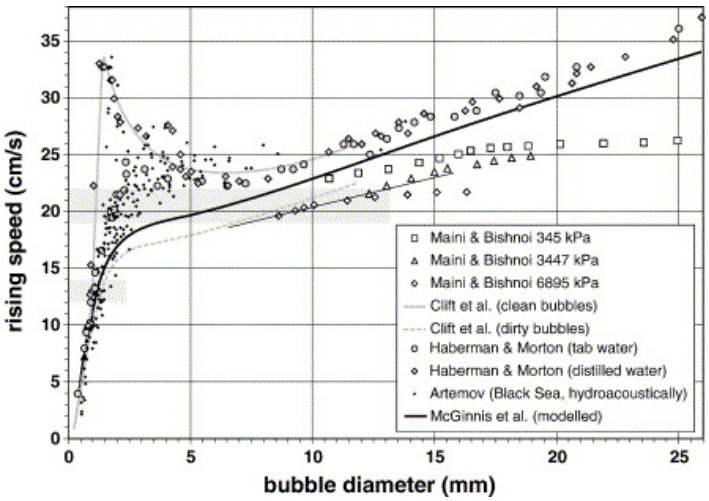


Figure 1.1.: Comparison of different mathematical models and measurements of bubbles. From [GAE+06]

composition. From this it follows, that a direct measurement of the bubble speed is required.

Bubbles can be observed by different means. In marine research the prevalent methods are either hydro-acoustical or optical. The focus of this thesis is on the optical methods and especially the algorithms used for the processing of acquired data. However because of their importance in past and current deep sea bubble research, first the hydro-acoustical methods are presented. In following part different optical systems are discussed, with a focus on camera based systems and the algorithms employed. An alternative method for the measurement of bubble plumes is to measure differences in the electrical conductivity of the water, but this method can, as [VF98] claims, be considered to be inferior to the former methods and is hence not described further.

1.1.1. Methods

Acoustic Measurement

Acoustic reflections from bubbles can be used to measure their size, speed and location. A sonar is used to emit an acoustical beam towards a suspected location of bubbles. The sound waves are reflected by the bubbles and recorded by a receiver.

The timing and phase of the signal can be used to calculate the distance and speed of the source of the reflections.

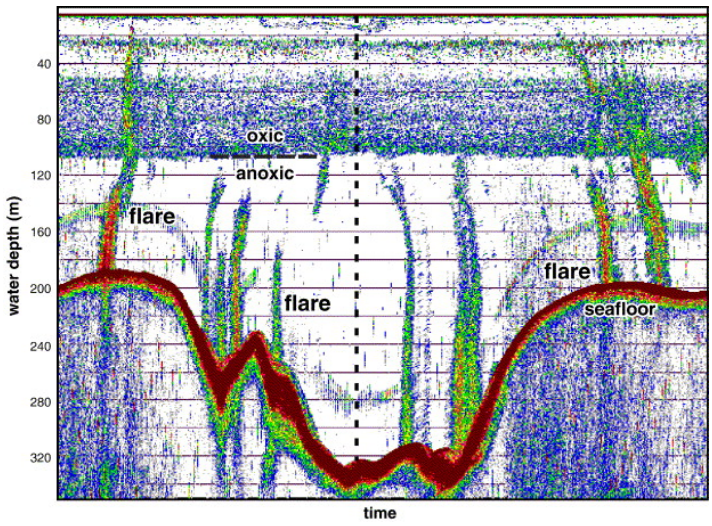


Figure 1.2.: Echogram of an exploration vehicle moving over depths between 190 m and 325 m. The recording is 22 min long. The flares show rising bubbles. The bending of the flares is attributed to currents and ship movement over time. Apart from these outside influences, the image was taken in ideal conditions. From [GAE+06].

With hydro-acoustic sensing large areas can be scanned for bubbles and later explored using other more accurate sensors, which can be carried on a ROV (remotely operated vehicle), a remote controlled underwater robot, as in [GAE+06]. Sophisticated research into marine seeps and their bubbles has been made using these techniques (e.g. [MGA+06]). The rising velocity can be determined by measuring the shift in frequency caused by the Doppler effect. In Figure 1.2 an echogram of an exploration vehicle moving above underwater seeps is shown. The image shows the high resolution and precision of current sonar systems. These observations were made while the vehicle is moving. Even small reflections beneath the track of the vehicle, including the seafloor profile and bubbles of different sizes, can be recorded. Such echograms are recorded in real time and allow the localization of gas seeps. The strength of acoustic sensing methods is their large range and the easy implementation. Accurate measurement of bubble size and gas volume with a

sonar alone is however difficult, , because even microscopic bubbles show a strong hydro-acoustic response [SvDP12]. In [GAE+06] flares which are not bubbles, but other rising sediment could not be easily distinguished from bubbles. This was solved by inspecting every potential source of bubbles using a ROV. The capabilities of single beam systems can be increased by using modern multi-beam sonars, which employ multiple beams. This has the advantage of higher accuracy as shown in [SvDP12], however the system developed in [SvDP12] is limited in range. Another disadvantage of hydro-acoustic measurement systems (e.g. [GAE+06]) is, that they assume that all bubbles are spheres, which means that the displayed accuracy of the data can be misleading. Therefore the measurement of the sonar should be verified with a direct method, for example with a camera.

Optical Measurement

In this section optical methods for bubble observation will be discussed. Optical systems can be categorized by illumination:

- X-ray
- Laser
- Visible light

X-ray illumination is best used for the observation of micro bubbles and was only tested in a laboratory setting (see [LK05]).
There are different kinds of laser based illuminations. One option is to use the laser for a coherent illumination and examine the decoherence of the back scatter [Buc94]. In [BB87] a well-defined volume of is illuminated by a laser generated pattern. Rising bubbles in this volume are identified by the disturbances they cause in this pattern. For this approach to be effective the granularity of the pattern has to be adjusted to the bubbles. It is ineffective if the bubbles are greater than $1\mathrm{cm}$ or occur in high density [AF95]. Another option is to use a laser to illuminate a horizontal plane in the water and observe the reflections of bubbles crossing this plane (see [MWZC11]). The camera in this case is oriented in the same direction as the light source. This is called front side illumination.

[MWZC11] is another example of a front side illumination bubble meter, but with visible light. Although front side illumination is theoretically well explored [MBD88],

generally back light illumination is preferred. The main reason is that even though the surface of a bubble has high reflectivity, as explored further in [LYX+06], by using front side illumination, the light is scattered and only a small fraction reaches the camera. The front illumination in [JC79] uses three pulse light sources to produce sufficient light.

Backlight illumination is the most practiced approach. There are many examples of bubble meters with diffuse back light illumination [LDLC03] , [LdLKHC03] , or [WD09b]. It delivers good illumination due to the light reaching the recording device directly. Nevertheless a strong illumination is helpful. In [LDLC03] a $1000W$ spotlight is used, other research features similar illumination devices [WD09b]. High frame rate cameras are commonly used in bubble meters. Bubble meters such as those used in [CB06] and [LDLC03] use a 1000 fps high speed camera.
In [LDLC03] experiments about the influence of the exposure time on the image quality were conducted. The result is that for the given bubble meter and a bubble size of 680μ an exposure time smaller than $1/250s$ is needed, which further reduces the available light. A light source with high intensity allows a low numerical aperture of the optics, which can significantly increase the depth of focus.

Camera based optical bubble meters require processing of the acquired imagery. Particularly in older designs (see [JC79]) the bubbles are extracted and counted manually. For the automatic extraction of the flow information from the acquired imagery bubble recognition algorithms are necessary.
In [CB06] small templates of previously identified bubbles are used for a bubble recognition algorithm. In the first step the templates are manually selected and stored in a database. The templates are identified in the camera image using principal component analysis and their position is improved using optimization. This method is verified by the authors in a laboratory setting and claimed to be in use for the research of heat transfer between surfaces and liquids. It remains unclear whether this approach would yield equally good results if illumination or bubble size were to change. Manually selecting the templates after every change in conditions like illumination seems impractical and it seems that the user has no direct feedback, whether the templates he selected are sufficient for high bubble recognition rates.
For the recognition of vapor bubbles in seething water [CB01] employs snakes (for an introduction to snakes see Section 2.1) , which are initialized with the boundaries of the binarized input image.

Another method presented in [GL97] uses a specialized Hugh transformation on the input image to locate the circular or ellipsoid shape of bubbles.
That bubble recognition is an active research field, is shown in [LHQ+13], which presents an new algorithm for extracting a bubble contour, published last year. An edge detector based method is presented in [TZSB10]. In a backlight illuminated bubble meter images are acquired with a high frame rate camera. The contours of the bubbles are extracted using the Canny edge detector. An ellipse is fitted in these contours. In this paper the method is tested in real conditions and the quality of the method is verified by a comparison with manually created ground truth data.

The quantification of the volume of the bubble flux using camera images is the goal of many works in bubble measurement. In [TZSB10], due to the monocular setup no depth information is available. To acquire volume information, it is assumed that the bubble depth equals bubble width. A similar approach is chosen in [LDLC03], but with smaller bubbles and spheres instead of ellipsoids. An alternative is used in [WD09a], in which a bubble model is employed in order to determine the volume of a bubble from its rising speed, buoyant force and drag force.

The method is tested in a laboratory tank, but this model it does implement parameters for bubbles from under water seeps, such as the methane concentration. The methane concentration is an important parameter, because as shown in [SvDRG+11] the bubble size is strongly dependant on it. For this thesis, the approach from [TZSB10] is chosen. This is a general method, which is applicable in deep sea and does not require several so far unknown input parameters such as [MGA+06],

Multiple cameras allow the reconstruction of the 3D shape of individual bubbles. Two cameras are employed in [BDZ+13]. The cameras are placed to observe the bubbles from perpendicular angles. First the bubbles are detected as ellipses from each camera, by using a combination of a threshold, Canny edge detector and the Hugh transformation. Afterwards an ellipsoid is determined from the ellipse parameters. In [CTMM02] perpendicular cameras with a strobe illumination are used together with an extra camera sensitive only to an additional laser illumination to generate a 3D reconstruction of rising bubbles, together with the fluid flow around rising bubbles.

2. Basic Concepts

2.1. Snakes

The term snake was coined by Kass in [KWT88] and describes a spline, which is adjusted by the snake method to fit a prior defined criterion. The spline or curve used in this context is called contour, because the most common use case is finding contours of objects in images. A contour segment can imply an image segmentation into the areas on, right, left, inside or outside of the contour. Consequently snakes are categorized as an image segmentation method.
Before a contour can be adjusted, a description of the contour is necessary to interpolate the signal to be segmented or adjusted. This description of a contour can either be explicit or implicit. The most influential explicit methods use a set of points or a parametric spline curve to describe the contour.

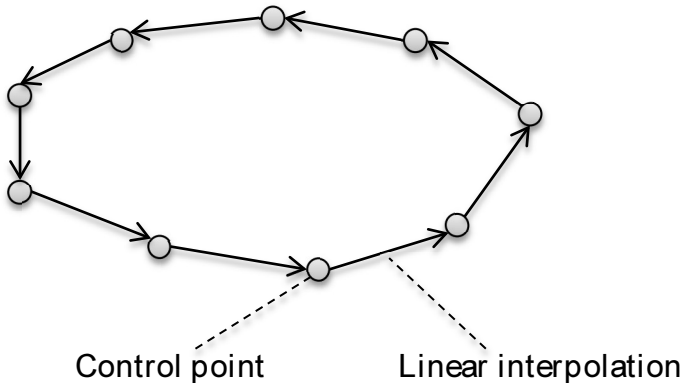


Figure 2.1.: Drawing of a snake consisting of control points and a linear interpolation in between them.

A set of points for the description of the snake spline is used in the original paper [KWT88], which coined the term snake for this concept. Using a set of points implies

piece-wise segments, which are linear in between control points. Such a snake with control points and the interpolation in between is shown in Figure 2.1.
Parametric splines, in this case b-splines, are used in b-snakes as in [MSMM90] and [MPG90].

In contrast implicit methods use level set to describe contours. Level sets are defined as levels of the same value in a characteristic of a higher dimension than the segmentation used. This difference in description distinguishes level-set methods from snakes, while snakes are also known as active contour models. In [PD00] an introduction to the topic is made, while [OF03] introduces active contours of higher dimension for the segmentation of surfaces from three dimensional data. They have a much higher computational complexity than other methods, but allow multi-region segmentation, as level set algorithms allow splitting and merging of regions. In this thesis active contours are understood, as an umbrella term for adjusting contours, while snakes (active contour models) and level-set methods are its representatives. This nomenclature is also used by Szeliski in [Sze11]. Be aware that not all older publications, e.g. [PD00] follow it.

This section focuses on the methods with an explicit description using a set of points, which are used in this thesis. In the snake algorithm a spline is adjusted to best fit prior defined characteristics. This adjustment can be seen as an optimization process, while the characteristics form an energy function. The snake moves from its initialization towards its final shape and position, which are chosen according to these characteristics. This slithering is the reason for the term snake (see [KWT88]). These characteristics are also called forces, as they are the driver of the motion of the snake during the optimization.

For every contour, an energy function E_{snake} is declared. This energy is used as a measure of merit of the contour. If the energy is minimal the contour fits the data best. The energy of the snake depends on its position and shape. Let $S : M \rightarrow P$ be a snake, with $M = [0, 1]$ and $P \subset \mathbb{R}^2$ all points on the snake. Let S_{final} the optimal configuration of the snake:

$$S_{final} = \arg\min(E_{Snake}(S)). \tag{2.1}$$

This energy can be described in two parts.

$$E_{snake}(S) = E_{int}(S) + E_{ext}(S). \tag{2.2}$$

The first part of the energy is irrespective of the actual image signal and depends only on the properties of the spline itself, it is called the "internal energy"E_{int}(see also [KWT88]). It incorporates the prior knowledge about the model of the object segmentation.

Continuity term describes the distance between control points of the contour. It is determined with the first derivate of the curve and has a shrinking effect on the snake. It can be compared to a rubber force compressing the contour.

Curvature term describes the curvature of the contour. It is determined with the second derivate of the curve and has a smoothing effect, penalizing against sharp corners. It can be compared to the properties of a thin metal plate, which cannot be bend in sharp angles as explained in [Sze11].

The continuity is weighted with factor $\alpha \in \mathbb{R}^+$, the curvature term with factor $\beta \in \mathbb{R}^+$.Let $c \in M$ be a position on this snake then the internal snake energy is defined by:

$$E_{int}(S) = \int_0^1 \alpha \|S'(c)\|^2 + \beta \|S''(c)\|^2 \, dc. \tag{2.3}$$

The external energy is used to describe the part of the energy, which depends on the image signal. It is weighted with parameter γ Given image $I : [0, x] \times [0, y] \to R^+$ with width $x \in Z^+$ and height $y \in Z^+$, it holds that:

$$E_{ext}(S) = \int_0^1 -\gamma \|\nabla I(S(c))\|^2 dc. \tag{2.4}$$

The external energy functions as a data term and attracts the snake towards strong gradients, while the continuity term and the curvature term function as regularization. Without influence from the external energy, which means without an image gradient on the snake contour, a snake will shrink until it encounters an image gradient.

2.2. CMA-ES - Optimization algorithm

Numerical non-linear optimization is about finding the argument for which a fitness function $f : x \rightarrow f(x)$ yields minimum value:

$$x = arg\,min(f(x)). \tag{2.5}$$

As an example, in Figure 2.2 a fitness function f with $arg\,min(f(x)) = 1.5$ is shown. The fitness function may be non-linear.
Several different search strategies for non- linear optimization are known. Typical numerical optimization algorithms begin with a starting point in argument space. In the following iterations the algorithm searches for a fitness function argument with lower value, until a convergence criterion is met. The minimum found can be a local minimum but not necessarily a global minimum. A classic method in numerical non-linear optimization is the gradient descent method [Noc06]. Its main principal is that with every iteration the current optimum is updated with a step in gradient direction. Another popular method is the Levenberg-Marquardt optimization algorithm [Mar63], which is a blend between the gradient descent method and the faster, but less robust Gauss-Newton algorithm.

CMA-ES is a different optimization algorithm [HK04]. It does not need a derivative of the fitness function. The ES stands for Evolution strategy it identifies CMA-ES as part as an evolutionary algorithm. Therefore certain terms for the concepts such as population are used in the description of the algorithm. The algorithm begins with a chosen starting point, the variances in each dimension, a population size and a termination criterion. Using a Gauss distribution with the initialized mean and the variances samples of the fitness function are evaluated. These samples (individuals) form the population and the chosen population size specifies its cardinality. Using the values of this population, the Covariance Matrix Adaption (CMA) updates the covariance matrices. Individuals which are inconsistent with the new covariance matrix are discarded. A new mean value is calculated.

Figure 2.3 shows an idealized CMA-ES optimization in the three steps. In the first step a) samples are taken according to the initialization covariance and mean. The green ellipse shows the 3σ distribution level determined by the covariance matrix and describes the distribution of the red samples. In step b) the covariance matrix

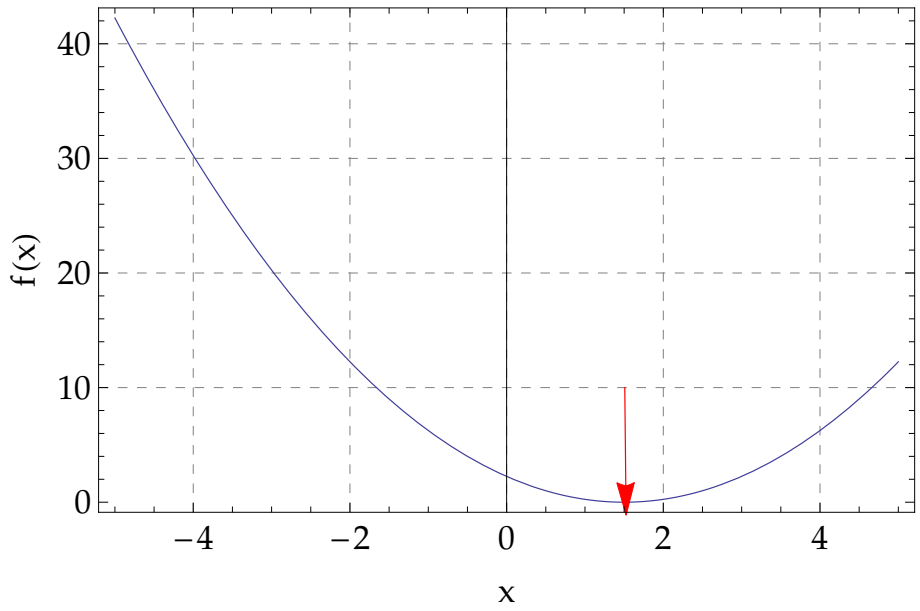
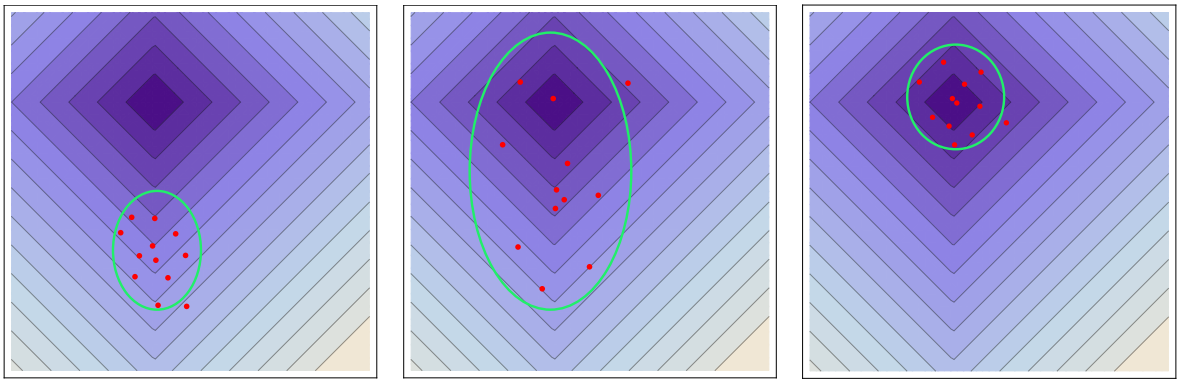


Figure 2.2.: Example for the argument of the minimum of an arbitrary function f. The $\arg\min$ of function f has the value 1.5 and it highlighted with a red arrow.



(a) Starting covariance (b) Covariance adaption (c) Final optimization state

Figure 2.3.: Idealized CMA-ES minimizing the absolute value function for which a contour plot is shown in the background. Darker hues of blue symbolize lower values of the target function. The 3 σ level of the variance determined by the covariance matrix is shown as a green ellipse and the individuals of the CMA-ES population as red dots.

and the mean are adapted in an attempt to localize the minimum. In step c) the minimum is found and the algorithm stops according to a predefined stop criterion. The argument minimum has been located as the coordinates of the sample with the lowest values.

2.3. Kalman Filter

The Kalman filter is an approach for obtaining the specifications of a linear dynamic system. It is a widely used signal processing algorithm and also has many applications in other fields of study, such as computer vision and particularly in tracking motions, as explained in [BMCM97]. The primary paper describing it from 1960 is [Kal60]. The formulas presented in this introduction are derived in [WB95]. The variable notation for the Kalman filter is as follows:

Variable	Description
x	state vector
u	control input vector
v	measurement noise, normally distributed with R
w	process noise, normally distributed with Q
P	estimated error covariance matrix
z	estimated measurement
A	model matrix
B	control input matrix
H	measurement model. It maps the Kalman state to new measurements.
K	gain of the system, determines how new measurements influence the system
R	measurement noise covariance
Q	process noise covariance

Capital letters denote matrix variables, small letters vector variables. Variables with subscript letters show time dependent states of the variable and variables marked with an apostrophe are updates.

The Kalman filter is a method to estimate the state of a random process at discrete points in time based on noise measurements. The process whose state is estimated is

described by:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_k. \tag{2.6}$$

In this equation x_k is the state vector at time step k, depending on the state in the last time step x_{k-1}. The transition matrix A describes the model of the process and the control input vector u_{k-1} together with matrix B describe how an optional control inputs influence the system. Note how the state x_k at time step k is only dependent on the last state x_{k-1} from time step $k-1$. This is why the Kalman filter can also be seen as a Markov process (see [MM77]). The new measurements z_k are described by:

$$z_k = Hx_k + v_k. \tag{2.7}$$

In this model the measurement vector z_k is assumed to be the result of the matrix H, which describes how the measurement is influenced by the state of the system, the state of the system x_k and the measurement noise v_k. The Kalman filter assumes that the process noise vector w_k is normally distributed with variance matrix Q, also known as the process noise covariance. The same applies for the measurement noise vector v_k, with variance matrix R, also known as measurement noise covariance. The goal of the Kalman filter is to find a good estimate of x_k.

The Kalman filter algorithm can be parted into two phases. The prediction phase in which x_k is predicted and the update phase in which a new measurement is incorporated into the Kalman filter to prepare a new prediction phase. This prediction is optimal in such a way, as it minimizes the quadratic prediction error (see [Kal60]).

Prediction A new prediction x'_k is generated by evaluating the model with the old state x_{k-1}:

$$x'_k = Ax_{k-1} + Bu_{k-1}. \tag{2.8}$$

The next covariances are estimated by

$$P'_k = AP_{k-1}A^T + Q. \tag{2.9}$$

Update After a measurement has been made, it can be used to refine the filter to improve the next prediction. This is accomplished in the update phase by evaluating

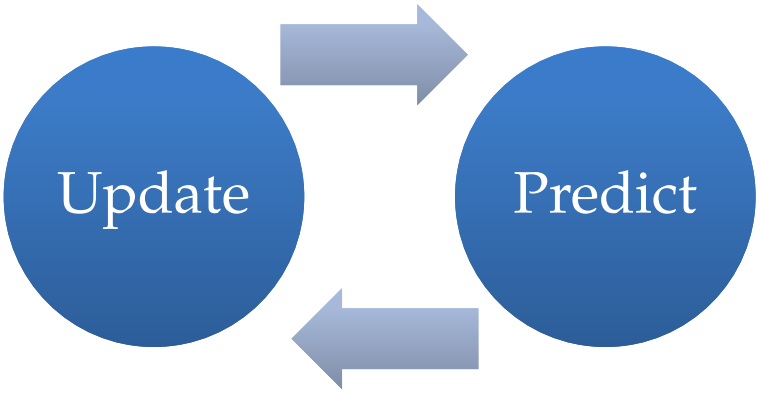


Figure 2.4.: Prediction and update phase of the Kalman filter are evaluated in turns.

the following equations, with z_k being the new measurement.

$$K_k = P'_k H^T (H P'_k H^T + R)^{-1} \tag{2.10}$$

$$x_k = x'_k + K_k(z_k - H x'_k) \tag{2.11}$$

$$P_k = (I - K_k H) P'_k \tag{2.12}$$

The x'_k and P'_k denote the state vector and the covariance matrix predicted in the prediction phase. The gain K gives a measure on how much the new measurement influences the system. The prediction phase and the update phase are evaluated in turns, as new predictions and measurements are made, see Figure 2.4.

3. Bubble Detection

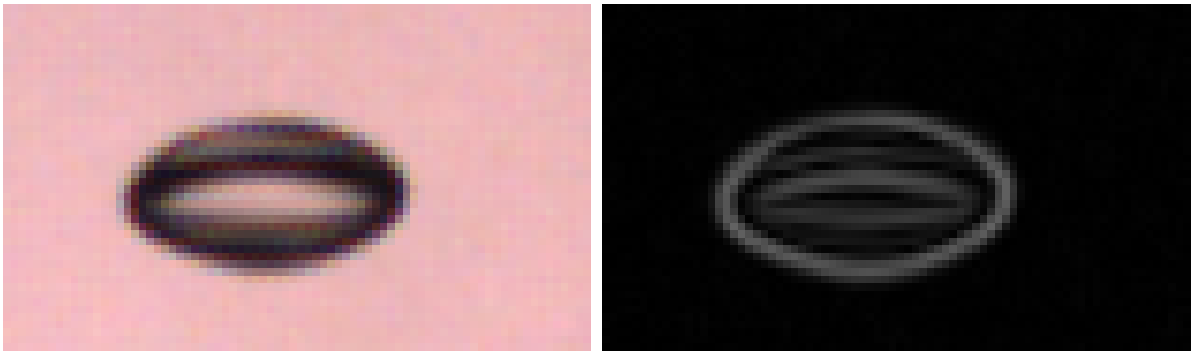
The most important component in a camera based bubble meter is locating the bubbles. An example of such a bubble image is shown in Figure 3.1a. This chapter presents different methods to locate and extract the shape of bubbles in an image. An overview over the methods presented in this chapter is depicted in Figure 3.2. They are the Canny edge method (see Section 3.1), the CMA-ES based method (see Section 3.4 and the snake based method (Section 3.3). Both the CMA-ES based methods and the snake based methods use a modified Canny edge method based initialization, described as coarse localization in Section 3.2).

3.1. Canny Edge Method

This algorithm is designed to be very similar to a algorithm used by [TZSB10]. The original implementation is based on proprietary Mathworks Matlab [1] functions. For this description and the following implementation, equivalent methods, like least squares ellipse fitting are used. The aim is to achieve comparable results.

First a Gaussian filter with kernel size 3x3 is employed to minimize potential noise, which may disturb the following processing steps. On the filtered image the Canny edge detector [Can86] is used. It extracts edges from the image. The Canny edge detector is a multi-step detector. The first step is finding the gradient intensity image using the Sobel filter. In this image the gradient strength and direction is determined and the direction is rounded to horizontal, vertical and diagonal directions. A non-maxima suppression filters the gradients depending on their direction. In the last step thresholds is applied, by accepting pixels with high gradient as edges, discarding those with low gradient and accepting those with an medium gradient depending on their neighbors. For more details about the Canny edge detector see [Can86].

[1] `http://www.mathworks.com/products/matlab/`



(a) Cropped image of a bubble with 78 pixels width and 46 pixels height.

(b) Sobel filtered bubble image. Brighter pixels show higher gradient values.

Figure 3.1.: Example images of a bubble. Characteristic are the dark rim and a bright spot inside. This is a segment of an image acquired with the Grasshopper camera. For the experimental setup see Section 5.1.

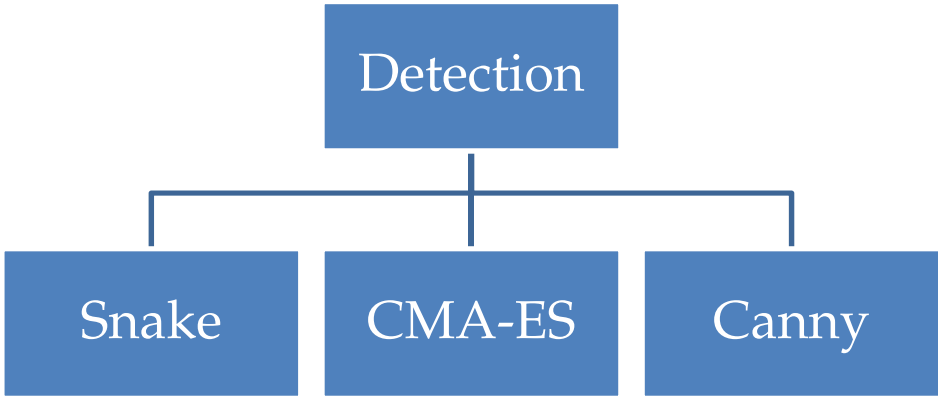


Figure 3.2.: Detection methods discussed in this chapter.

The connected components of these edges are determined and with these the detected edges are extracted from the image. Into every edge an optimal ellipse is fitted using the least squares method [FF95].

3.2. Coarse Localization

The goal of the coarse localization is to localize all bubbles in an image. The exact shape of the bubbles is not detected, but a bounding box or rotated rectangle which circumscribes each bubble. The exact shape of the bubble is left for subsequent methods, like the snake method in Section 3.3.
While the canny edge detector reliably detects the bounds of bubbles, due to the irregular shape of the bubbles, the contour detected by the canny edge detector is not necessarily closed. This is because some parts of a bubble contour may not have a strong gradient.

As an improvement over the Canny edge method only the outermost connected edges are considered. Nested edges may occur, because the dark rim around bubbles leads to two edges, the edge on the outside of the rim and the edge between the rim and the brighter inside. These edges can be connected, if the dark bubble is narrow. To establish the outer bounds a bounding box in the form of a rotated rectangle is traced around every contour.

Against false detections, heuristics such as a minimum bounding box size of 3 pixel and a maximum size 100 pixel are employed.

The snake method locates contours with high gradient values near its starting position. To ensure that the outermost contour is chosen this bounding box is expanded by 120%, if it is used for further processing with the snake method.

3.3. Snake Bubble Detection Algorithm

The general snake concept has been introduced in Section 2.1. This section presents the implementation and different improvements of this snake algorithm.
The snake algorithm by Kaas et. al. is implemented in [KWT88] by optimizing the positions of all snake control points at once. In [WS92] a different approach is presented, the greedy snake algorithm. It is designed to be much faster than the original implementation in [KWT88]. The snake algorithms in this thesis are implemented in

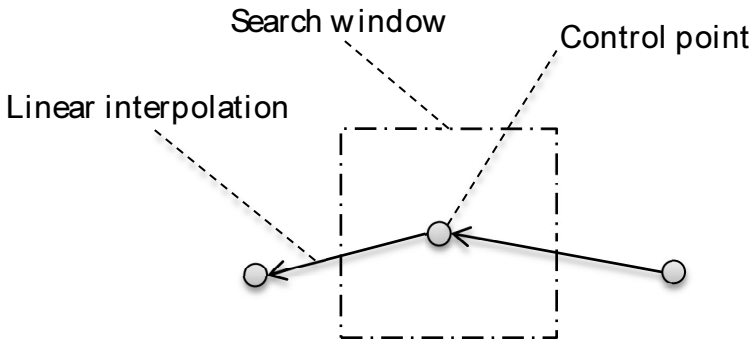
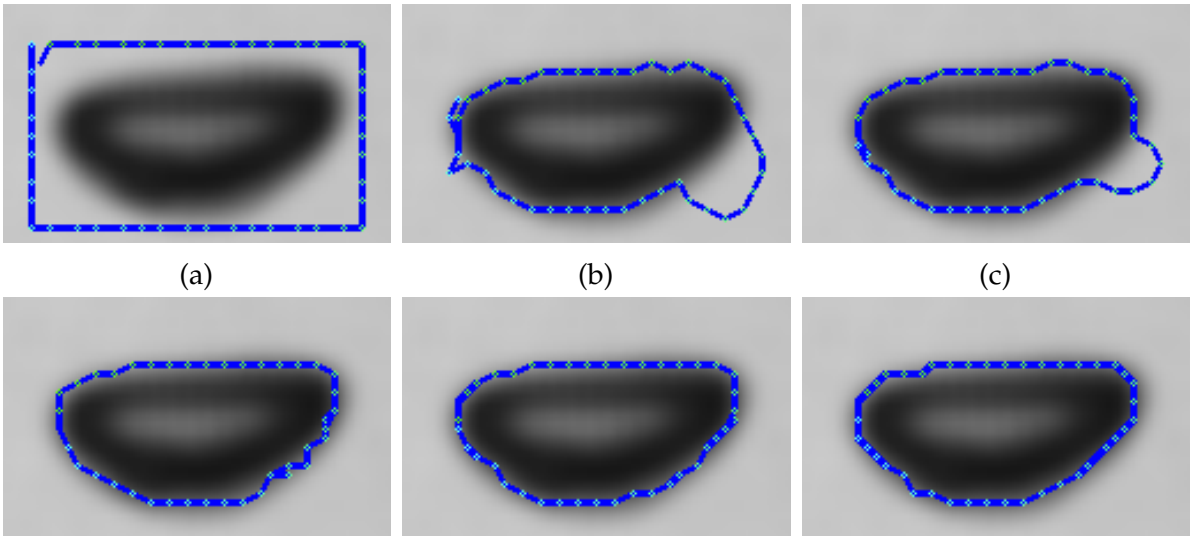


Figure 3.3.: Drawing of the control points of a snake and the search window, in which the next control point position with minimal energy is located.

this fashion.

The snake contour is initialized with the bounding box from the coarse localization. In each iteration of the snake algorithm, the control points are handled consecutively and separately. For every control point, the potential energies are calculated for a permissible range of motion, also called search window. Note that this energy, calculated from the internal snake energy, which is composed of continuity and curvature aspects, according to Equation 2.3 and the external snake energy gradient (see Equation 2.4), which incorporates the image gradient, is valid only for this point. A total snake energy using the integral or sum over the entire snake is not calculated. The search window around a control point of a snake is shown in Figure 3.3. From this search window the potential new position with the lowest energy is chosen and the control point is updated to this position. After all control points have been optimized, the stop criterion is checked, and if necessary the next round of optimizing is started. This optimization results in the movement of the snake into a better fit. An example of the different optimization steps of a snake on the image of a bubble is shown in Figure 3.4.
The stop criterion has two components. The optimization stops if the snake is converged. This is determined by checking if in the current iteration no control point positions were changed since the last iteration. The second stop condition is a fixed limit in iterations. This means that if a fixed number of iterations is reached, the algorithm stops. It is assumed that a good enough result is found within this number of iterations, even if the algorithm does not terminate.



(a) (b) (c)

(d) (e) (f)

Figure 3.4.: Subfigures (a) to (f) show every second step in the optimization process of the snake algorithm, from initialization to termination. The control points are depicted in green, while the linear interpolation in between them is shown in blue.

Another aspect of the snake algorithm following [WS92] is that before the energies in the search window are calculated as their weighted sum, the continuity and curvature, which are the internal snake energy (see Equation (2.3) and the gradient values, which form the external snake energy (see Equation (2.4), are normalized with their minimum and maximum values within the search window. The effect is that the optimization becomes more robust, as the balance of the three components is maintained in non-uniform conditions.

The outer contour of the bubbles is the goal of the snake optimization process. The snake has the property that it is shrinking if regions with high gradient values are outside of the search window. This means that without external (image dependent) influences the snake is shrinking. Because of this behavior, the snake is initialized outside of the bubble. During the optimization the snake shrinks to fit the bubble.

3.3.1. Unifying Redundant Control Points

If a constant number of control points is assumed, the distance between the control points in final snake is smaller than in the initialization. As the control points are subject to the discrete pixel locations of the image, in the course of the optimization the control points, which at the initialization were properly pitched, share the same position. The continuity term of the snake energy function requires the calculation of the derivative, by means of calculating the distance between control points. This calculation becomes invalid if the points have identical coordinates. Similar problems occur for the calculation of the curvature term of adjacent points.
The solution to this situation is unifying control points, which share position. These control points are redundant and can be removed, because they hold no additional information about the contour of the bubble. Using the removal of redundant points, strongly contracting snakes are safe to use.

There are different key factors to consider for choosing the appropriate spacing of control points. The first factor is runtime, because the snake algorithm scales linear with the number of control points. The second factor is accuracy. The more control points are used, the finer grained can the final snake match the bubble. Nevertheless for a contracting snake, using very closely pitched or even touching control points in the initialization offers no advantages. Many will be removed anyway in later phases of the optimization, because of the redundant point removal strategy. This leads to the refinement method presented in Section 3.3.3.

3.3.2. Gradient Direction Snake

In this section a novel improvement of the snake algorithm is presented. The classic snake algorithm as described in the previous Section has no concept of outside or inside of a contour. A gradient on the inside of the dark rim of a bubble is weighted just like the gradient on the outside.

The goal of this extension is to make the localization of the bubble contour more robust against the detection of inner edges.

On a typical bubble image, the bubbles are delimited with a dark rim. See Figure 5.4 for an example. On the outside bounds of this rim the image intensity values get increasingly brighter, until the background intensity is reached. These intensity variations determine the image gradient.

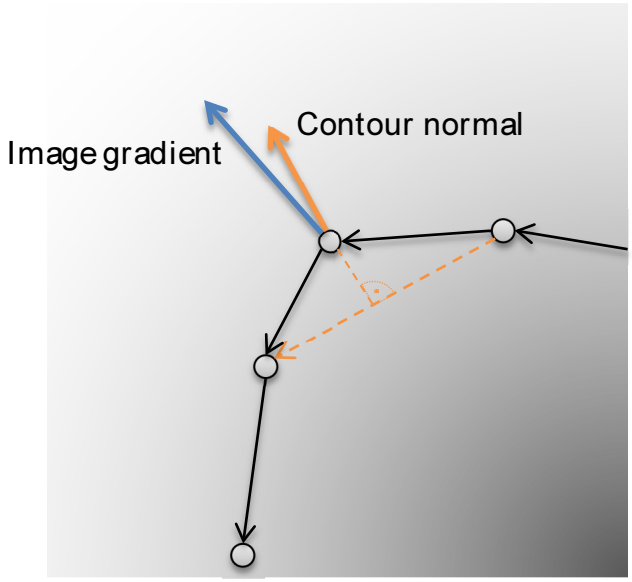


Figure 3.5.: Drawing of a snake with control points and the linear interpolation. In the background different image intensities are shown, which in a certain control point result in an image gradient vector, depicted in blue. The contour normal of this control point and its calculation scheme with the normal to the difference between the adjacent control points is shown in orange. The coincidence of the directions of these vectors is used as an optimization criterion in the gradient direction snake.

As visible on these bubbles, the gradient on the outer rim of a bubble always points outwards. As an extension of the snake method a novel weighting term is introduced, using this fact. In Figure 3.5 the direction of the image gradient and the contour normal in a snake control point are illustrated. To weight this effect a term, which is sensitive to the direction of the gradient on the contour is derived in the following.

With x being the horizontal component and y the vertical component, the normal at position c of Snake S is:

$$n(S(c)) = \begin{pmatrix} \frac{-dS(c)}{dy} \\ \frac{dS(c)}{dx} \end{pmatrix}. \tag{3.1}$$

For the normalized contour normal $\widehat{n}$, it follows that:

$$\widehat{n} = \frac{n}{\|n\|}. \tag{3.2}$$

At position c, the dot product is applied to the normalized image gradient and the outwards normal of the contour. If they point in the same direction, the result is positive and if they point in different directions it is negative. The result is scaled with the squared absolute value of the image gradient and is similar to E_{ext} weighted with parameter γ. For the following applies:

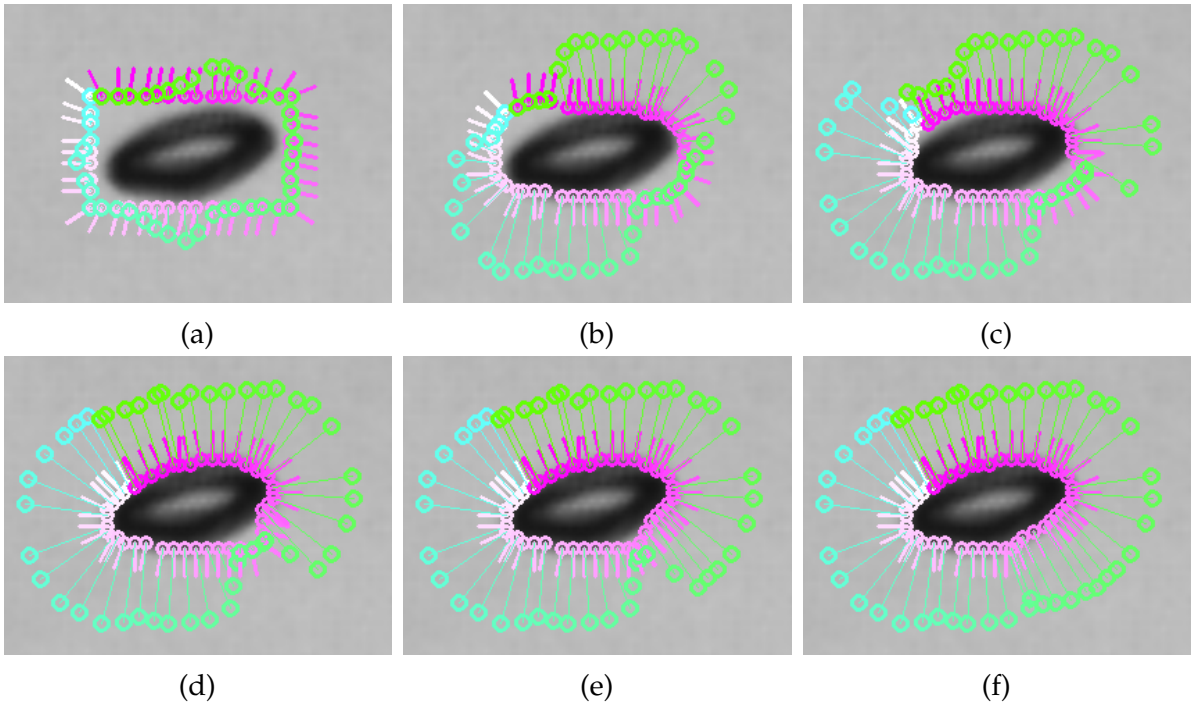
$$E[k] = \int_0^1 -\gamma \left\langle \frac{\nabla I(S(c))}{\|\nabla I(S(c))\|}, \widehat{n} \right\rangle \|\nabla I(S(c))\|^2 \, dc. \tag{3.3}$$

Different steps of the snake algorithm with the novel term, detecting the shape of a bubble, are shown in Figure 3.6 with the image gradient vector depicted in green hues and contour normal vector in lilac hues. They illustrate the effectiveness of the novel term particularly for the detection of bubbles.

In the implementation of the algorithm, derivatives are calculated by using finite difference between control points. While using both differences to calculate the derivative between the previous to the current and from the current to the next is more accurate, the derivative can be approximated with the difference between the next and the previous control point. This approximation is possible, because a bubble contour is forced to be smooth by the curvature term (see Section 2.1). The approximation is also reasonable, because if the current point is not involved in the calculation of the normal, it does not have to be recalculated for every potential point in the search window, the innermost snake algorithm loop. This saves calculation time. In Figure 3.5 this approximation technique for the contour normal is illustrated by the dashed orange lines.

For different values of n and $\nabla S(c)$, the dot product values change with the cosine function. The sensitivity of the cosine function is not linear and is lowest for values around -1 and 1. To achieve an accurate positioning of the snake on the gradient the values should be around 1, which is in the least sensitive value range.

Using the sum of the normal and gradient direction vectors, instead of the dot product, has the advantage of gaining linear sensitivity to angular difference between n and $\nabla S(c)$ over the $cosine$ and lower computational complexity. The difference



(a) (b) (c)

(d) (e) (f)

Figure 3.6.: Subfigures (a) to (f) show every third step in the optimization process of the snake algorithm with directional gradient, from initialization to termination. Lilac hues show the normal of the contour, green hues show the direction and magnitude of the gradient indicated by the length of the greenish indicators, which is scaled by factor 0.16.

is then scaled by 0.5 to achieve values, which are not scaled, relative to the original snake algorithm, for coinciding vectors:

$$E[k] = \int_0^1 -0.5\,\gamma \left\| \frac{\nabla I(S(c))}{\|\nabla I(S(c))\|} + \hat{n} \right\| \|\nabla I(S(c))\|^2 dc. \tag{3.4}$$

3.3.3. Refinement Method

A snake is represented by a sequence of control points. Besides the removal of redundant points (see Section 3.3.1) the number of control points is set with the initialization of the algorithm. The goal of this extension is to make the snake algorithm faster. The time complexity of the snake algorithm is linear with number of control points. Using fewer control points, means a faster algorithm, but a lower accuracy of the contour. The idea of this extension is to use multiple rounds of the snake algorithm with different numbers of control points, to overcome this limitation. This should not be confused with a multiple scale approach.
Using multiple image scales in combination with the snake algorithm is a well-established technique (see e.g. [KWT88]). The goal of using different scales is to make the snake more robust against converging to unwanted local minima. Local minima, which generally mean smaller structures, vanish if a coarse scale is used.
This effect is not wanted for the detection of bubbles. The snake is already initialized directly outside of the bubble. A coarse image scale might even be harmful, as it can allow a snake to overcome the outer boundary of the bubble, which it is meant to converge to.
Therefore the input image and evaluation of the snakes remain the same, but the spacing of the points are varied. For this extension a two-step approach is proposed. In the first step a coarse resolution of the snake is used. The snake converges to the bubble contour. In the second step the control points of this snake are interpolated to a finer resolution and again the snake converges to the bubble contour, this time with fewer iterations than in the first step, as the points are already nearly in the correct position.
The advantage lies in the lower control point count in the first step and the faster convergence in the later.

3.4. Ellipse Fitting with CMA-ES

This section is about the detection of bubble contours using the CMA-ES optimization algorithm described in Section 2.2.

In the previous section a bubble is approximated with a snake. The shape of the snake is controlled by the continuity and curvature terms, but only on a local per control point basis. Therefore the snake is free to form any shape, as long as the energy of all control points in that configuration is minimal in its environment. In this section a different approach is chosen by introducing a bubble-wise shape restriction based optimization technique. The general idea is, that this shape prior is initialized with the roughly correct position and shape and is then fitted with an optimization algorithm to resemble the bubble.

Looking at images of bubbles (see Figure 5.4), it is clear that the most direct method of approximating a bubble would be a circle. A circle has three degrees of freedom, the position in vertical and horizontal coordinates and the radius. However, only small bubbles can be accurately approximated by a circle. More flexible than a circle is an ellipse or a rotated ellipse. A rotated ellipse is used to approximate the contour, because it offers enough flexibility to fit most bubble contours, but has only five degrees of freedom. This results in the advantage of a high stability of the optimization, because with the less parameters the optimization becomes more robust. The position in horizontal and vertical coordinates, p_x and p_y, the horizontal and vertical axes, a and b, and the orientation of the ellipse the ellipse α are chosen as parameters. They describe the position, orientation and size of an ellipse, which are fitted to the shape of a bubble. An ellipse together with the optimization parameters is shown in Figure 3.7. The parameter vector for an ellipse E is

$$E = \begin{pmatrix} p_x \\ p_y \\ a \\ b \\ \alpha \end{pmatrix}. \tag{3.5}$$

This parameter vector is chosen as the target of the optimization algorithm, which varies these values until the fit is good enough or a set number of optimization steps

have been performed, as defined in a stop criterion.

For the optimization algorithm, the ellipse values need to be initialized with the coarse position and shape of the bubble, so that the optimization is able to converge the ellipse to fit the bubble well. In this thesis the parameter vectors are initialized with the results from the coarse localization as described in Section 3.2.

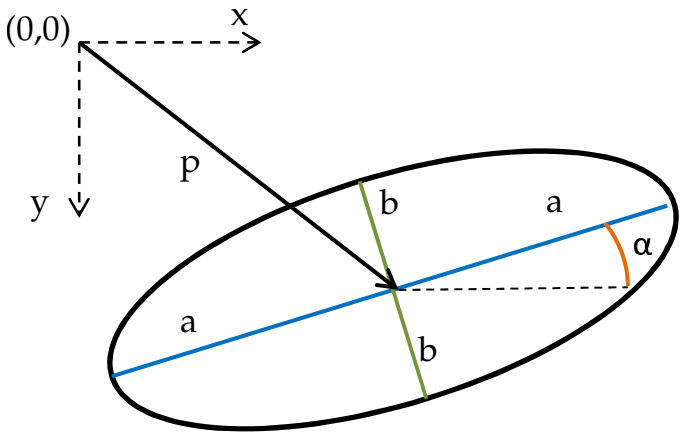


Figure 3.7.: Ellipse with optimization parameters. a is the horizontal axis of the ellipse, b the vertical axis, α the rotation of the ellipse, and p with components p_x and p_y the position vector.

The scope of the CMA-ES optimization, which defines how local or global the values of the parameter vector are varied, is depends on the starting standard deviation of the optimization parameters. The search must not be too narrow as otherwise the best fit might be missed, but also not too global as the targeted bubble in a per image perspective is a local minimum. Being able to control the optimization process is an advantage of CMA-ES over other optimization algorithms. Moreover CMA-ES is derivative free, so no derivatives need to be calculated and all necessary parameters like the population size and a termination criterion can be set directly. Therefore, despite being generally slower than other optimization algorithms for problems with less than 10 parameters [AH12], the CMA-ES optimization algorithm is selected as the optimization method of choice.

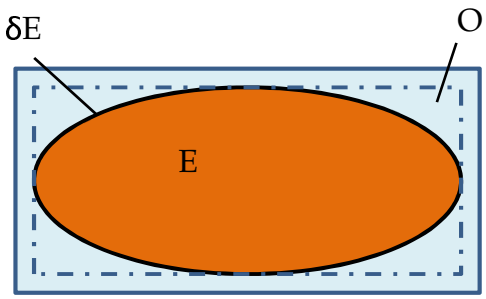


Figure 3.8.: The definition of different regions around an ellipse for CMA-ES fitness functions. The ellipse E is shown in orange, with its boundary ∂E in black. The circumscribing rectangle is shown in dashed blue and the expanded rectangle in blue. The area inside the expanded rectangle, without the ellipse, O is shown in light blue.

3.4.1. Fitness Functions

The previous section explained the choice of parameters and the choice of an ellipse as an approximation for the bubbles. The fitness function is the basis of the optimization. It defines what is considered a good fit and which conditions should be penalized. This section discusses the fitness functions used for the CMA-ES optimization. The optimization ellipse together with the variables used is shown in Figure 3.8.

Edge Gradient Function The two most distinguished features of a bubble are the dark rim or dark area inside it and its gradient on the outside edge. A straight forward approach which gets close to the snake approach, is to choose the high gradient on the edge of the contour as the deciding element.
Let $E \subset I$ be an open set of points forming an ellipse with ∂E the boundary of the ellipse and $p \in \partial E$ be a point on ∂E, the boundary gradient fitness function is:

$$f(E) = \frac{|\partial E|}{\sum_{p \in \partial E} (\|\nabla I(p)\|^2)}. \tag{3.6}$$

This evaluates the squared gradient per boundary point, inverted so that higher gradient values lead to lower fitness function values. Penalty terms against ellipses outside of the image or against collapsed ellipses are not displayed.

Edge Intensity Function Using the image intensity as a measure instead of the gradient, leads to another fitness function. It minimizes the intensity per ellipse edge point.

$$f(E) = \frac{\sum_{p \in \partial E} I(p)}{|\partial E|}. \tag{3.7}$$

A boundary based fitness function can be evaluated very fast, compared to a fitness function examining the entire ellipse, because the number of evaluation points is lower, however information such as a darker mean intensity inside of a bubble compared to the background is not used. Another disadvantage of this boundary based fitness functions is, that they are limited in its application to bubbles, whose shape is very near to that of an ellipses.

Threshold Function For a fitness function which uses the complete bubble, a rotated rectangle circumscribing the ellipse is chosen as the outside environment O. This fitness function uses the dark rim and other possible darker areas inside the ellipse. This direct approach introduces a threshold t and optimizes the number of dark pixels inside the ellipses relative to its area.

$$f_{thresh} = \frac{|p \in E : I(p) < t|}{|E|} \tag{3.8}$$

This function is equivalent to binarizing the image beforehand and counting the darker pixels in the ellipse. It has the disadvantage of introducing an arbitrary threshold, which would have to be updated by the user in case of changing conditions. Therefore a fitness function without a threshold is preferred, as it is independent of the background intensity and does not need user input.

Difference of Means Function The following fitness function uses the area of the ellipse and needs no user input. The goal is the best separation of the sum of image intensities between a circumscribed rectangle and the ellipse E, see Figure 3.8. The rectangle is expanded by 5% in each direction, for the function to also be sensitive in the direction of the axes of the ellipse. The area inside the rectangle, but not in the ellipse is defined as outside O. As a measure of the intensity and to eliminate possible bias from the points inside the ellipse or outside of it, the mean intensity

values inside μ_E and outside of the ellipse μ_O are calculated:

$$\mu_O = \frac{\sum_{x \in O} I(x)}{|O|} \tag{3.9}$$

and

$$\mu_E = \frac{\sum_{x \in E} I(x)}{|E|}. \tag{3.10}$$

For the separation of mean intensity values, the fitness function $f_\mu(E)$ optimizes the best difference in image intensity mean values, between inside of the ellipse and its surroundings:

$$f_\mu(E) = \frac{\mu_E}{\mu_O + \mu_E}. \tag{3.11}$$

Average Difference from Mean Function Another approach, similar to the calculation of the statistical variance and Chan-Vese term [OF03], calculates the best separation between inside and outside sets with the most uniform image intensity distributions:

$$f_s(E) = \frac{\sum_{x \in E} |\mu_E - I(x)\|}{|E|} + \frac{\sum_{x \in O} \|\mu_O - I(x)|}{|O|}. \tag{3.12}$$

Note that this function needs a good initialization, so that the inner and outer regions are already at least slightly separated.

Combined Difference Function Fitness function f_μ optimizes the separation of mean intensities, while function f_s optimizes the uniformity of these regions. For a good ellipse fit of a bubble, both criteria should be achieved, therefore function f_c is introduced, which combines both functions. This combination is also advantageous, because the mean intensity values calculated in f_μ fitness function are also necessary in function f_s. This means they can be combined without adding significantly more

computations. The fitness function f_c is calculated as follows:

$$f_c(E) = f_\mu(E) + f_s(E). \tag{3.13}$$

Substituting f_μ and f_s leads to:

$$f_c(E) = \frac{\mu_E}{\mu_O + \mu_E} + \frac{\sum\limits_{x \in E} |\mu_E - I(x)|}{|E|} + \frac{\sum\limits_{x \in O} |\mu_O - I(x)|}{|O|}. \tag{3.14}$$

As evident from this formula, the mean values μ_O and μ_E are needed for the evaluation of all parts of f_c. Furthermore additional weighting factors can be used to fine-tune the influence of each part of this function.

4. Bubble Tracking

To observe the size of bubbles over a longer image series and to measure the speed of the bubbles, it is necessary to track the bubbles.
The different methods for bubble tracking, which are presented in this chapter, are shown in Figure 4.1. The mean velocity approach is introduced in following section and the individual bubble tracking with a best matching approach together with Kalman filter prediction will be described in Section 4.2.2.

For the tracking of bubbles, the center of mass of the bubble contour is used as a reference point. This center of mass is henceforth used as the position of the bubble. The center of mass is calculated from the moments of the bubble contour M_{00} , M_{01} and M_{10}. For details on how to calculate the moments of a contour see [YA96]. Then horizontal center of mass c_x is calculated by:

$$c_x = \frac{M_{01}}{M_{00}} \tag{4.1}$$

and the vertical center of mass c_y by:

$$c_y = \frac{M_{10}}{M_{00}} \tag{4.2}$$

4.1. Mean Velocity Approach

Observing the movement of bubbles between two frames one can see that although the bubbles may not all move with the same speed, a mean speed can still be computed. In the following, the steps to determine the mean speed are described.

If a pair of bubble images in their corresponding frames shows the same bubble, then their distance is a valid estimation of the movement of this bubble between these frames. As the frame rate is constant, a bubble velocity estimation is possible. The vertical components of the distances in pixels between all bubbles positions in

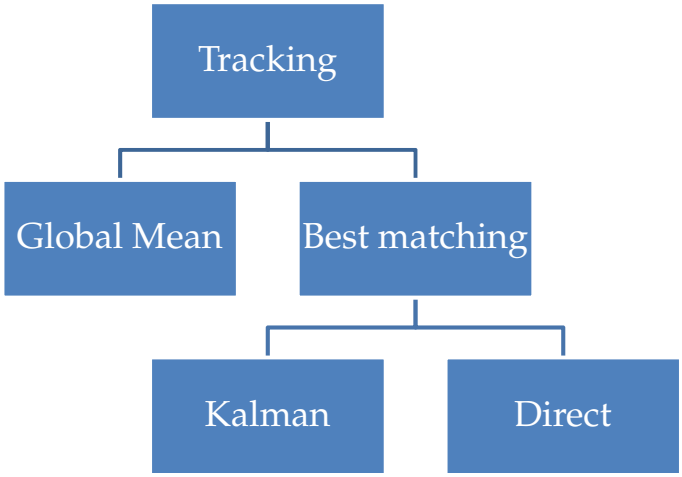


Figure 4.1.: Overview over the different tracking methods presented in this thesis.

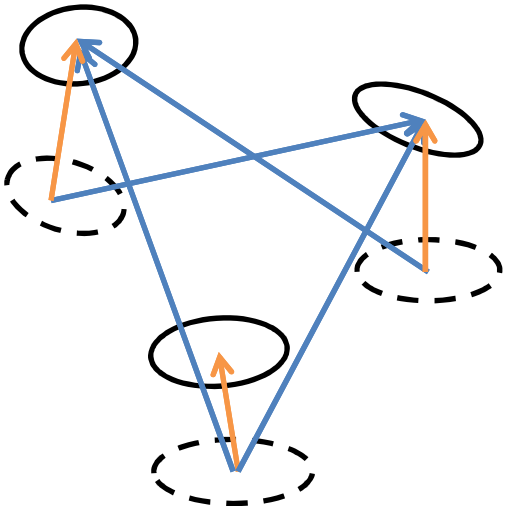


Figure 4.2.: Drawing illustrating the measurements of the mean velocity approach. Valid measurements are shown in orange and invalid measurement in blue. The current bubble positions are depicted with black ellipses and the bubbles positions from last frame with dashed ellipses.

one frame to the position of all bubbles above itself in the next frame are measured. A graph of these measurements is shown in Figure 4.3. A valid and correct measurement is defined as the measurement between the images of the same bubble. They are illustrated as orange connections in Figure 4.2. It is likely, that for every rising bubble the valid measurement is part of all these measurements made. If bubbles break apart or unify in between frames, it means that more than one distance measurement to or from a bubble can be valid. However a valid measurement may not always be possible, because of reasons such as the finite field of view or erroneous detections of bubbles. The next step is to distinguish the valid velocity estimations from invalid estimations.

If the distance values were continuous, the equivalent of a normalized histogram would be the probability density function. Let P be the probability density function of distance values x and the valid measurement x_{valid}. Then the valid measurement is:

$$x_{valid} = \max(P(x)). \tag{4.3}$$

A maximum can the determined by:

$$0 = \nabla(P(x_{valid})) \tag{4.4}$$

$$0 \neq \nabla^2(P(x_{valid})). \tag{4.5}$$

More than one maximum is possible, because in a bubble stream of equidistant bubbles not only the valid measurements show results of low variance, but also the measurements to respectively the next bubble. Hence it is important to use the first maximum (except if a camera with a very low frame rate is used).

Because the distance values are not continuous, a histogram H is generated. The distances are sorted in an ascending order and parted into bins. These bins are about ten percent of the maximum allowed distance wide. Counting the number of entries per bin results in an histogram. For all bubbles there is at least one valid measurement (for exception see the paragraph above) and all valid measurements vary only slightly compared to non valid measurements. It follows that the maximum in the histogram shows the valid measurements, under the assumption that the bubble speed distribution is not multi-variant. This means that bubble velocities are distributed around a common mean value. This assumption will be verified in the results Section 5.

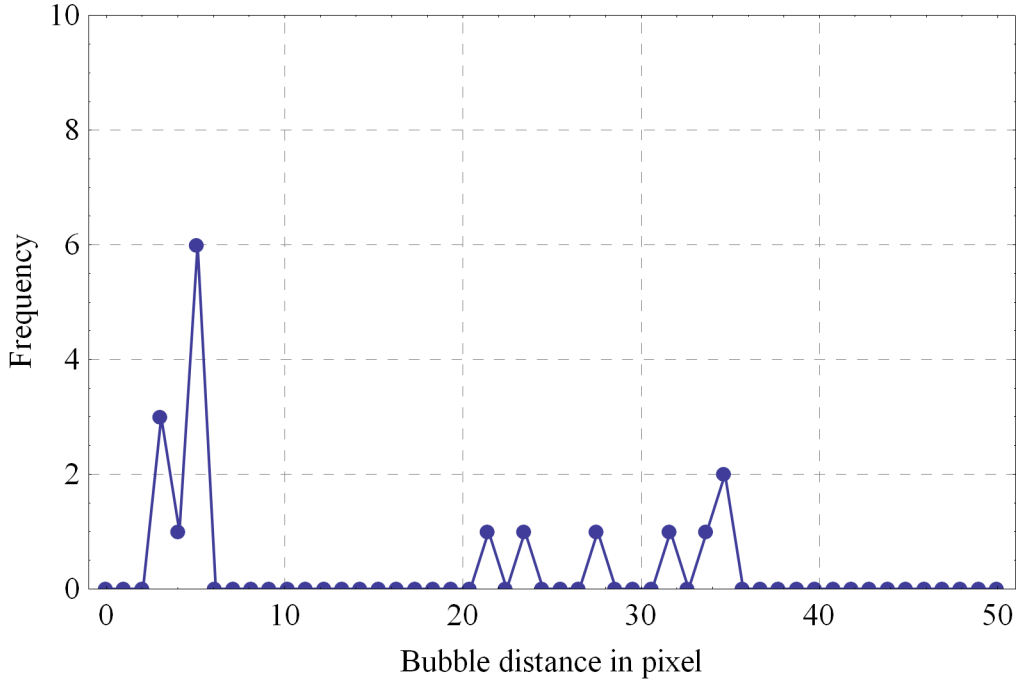


Figure 4.3.: Frequency of rounded vertical upwards distances measured between bubble positions from two consecutive frames. The data is generated from a image series acquired with the GoPro camera at 240 fps. An example of such an image is shown in Figure 5.4b.

The histogram is cropped to show a realistic range of possible bubble velocities, across the maximum allowed distance. This value is generally set to twice the value of the expected result. Note that in the illustrative figures in this section, the histograms are cropped at a much higher value to show the higher maxima.

If the valid measurements are distributed with a mean near the bounds of a histogram bin, the maximum in the histogram may not be as prominent as possible, because the valid measurements would be divided in two bins. Thus a second histogram is generated, in which the bounds of all bins are half a bin size shifted. Both histograms are shown in Figure 4.4. Values, which in the first histogram are distributed around the fringe of a bin are now in the center of one. The best maximum of both histograms is chosen, in case of Figure 4.4, the bin from the red histogram

with frequency 10. The mean velocity is determined, by calculating the mean of all values within this bin.

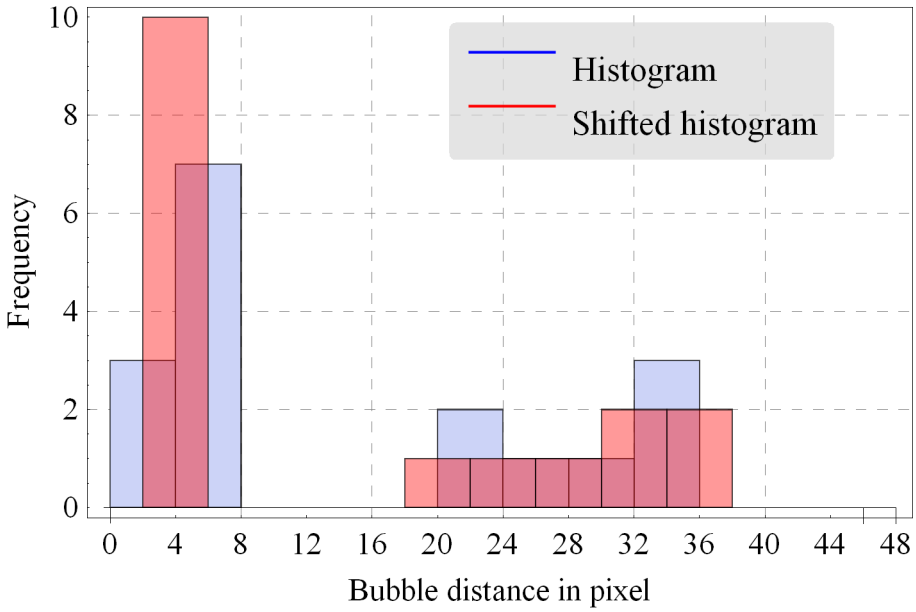


Figure 4.4.: Figure with overlapping histograms in red and blue of the same data as Figure 4.3. The x-axis values show the measured upwards distances between the bubbles. In the blue histogram the valid measurements are split into two bins. Therefore the maximum is not as prominent as in the shifted red histogram, which shows better results.

This mean velocity measure is not accurate for every individual bubble, but the goal is a good measure of the global mean ascending speed of all bubbles. Using the mean speed of the bubbles for further processing, like estimating the mean flux, may not be as accurate as measuring the speed of every bubble, nevertheless measuring the mean speed can be accurate enough for some applications.
If the speed of every bubble has to be measured, the speed measured in the current frame has to be recorded together with the speed of the corresponding bubble contour from the last frame. How to find these correspondences will be the subject of the following sections. The primary advantage of the mean speed approach is that no such information is necessary. In the course of the algorithm only the bubble positions from the last frames have to be saved. Accurate data about the speed distribution in

bubbles and an assessment about the usability of the mean speed value can be found in Section 5.4.

4.2. Individual Bubble Tracking

4.2.1. Kalman Snake

The snake algorithm as described in Section 3.3 supports tracking, as first described in [KWT88]. A snake outside of a contour gradually moves in the course of its iterations onto the contour. This motion can be used for tracking.In the following frame the snake is set to a position and shape identical to the last frame and using its contour fitting iterations, moves towards the position of the contour in the next frame. This type of snake motion tracking, while easy to implement is only feasible for small motion disparities between frames. If the motion is greater than half the size of the entire contour, it becomes impossible to use this type of tracking, because for both sides of the previous contour, one side of the next one is the closest next contour and therefore a local minimum in the snake optimization. This makes this type of tracking unsuitable for tracking of fast rising bubbles.
This is the motivation for Kalman snakes. A Kalman snake algorithm as described in [Sze11] employs a Kalman filter on the snake. The intention is to lower the motion disparity the snake algorithm has to optimize across, by using a Kalman prediction algorithm and then moving the snake in the next frame with all its control points.

Thus Kalman snakes allow the detection and tracking of individual bubbles. However applied to a bubble stream this technique has strong disadvantages to another method presented here.
Between the first frame in which the bubble has been detected and the second time it is identified the prediction delivers only values based on the initialization speed of the Kalman snake. Although a good value for vertical motion can be estimated from the observation of other bubbles, the horizontal motion of bubbles is at this stage not incorporated into the Kalman filter system. The Kalman snake prediction must be precise enough for the snake algorithm to locate the bubble in every frame, otherwise the snake algorithm will fail to converge to the correct contour. Such an error critically interrupts the tracking process. The necessary precision of the prediction together with the horizontal motion makes the Kalman snake unreliable in this context.
Another challenge concerning Kalman snakes is that for the localization of bubbles a

detection is needed to determine whether a bubble is already tracked using a Kalman snake. In this thesis a different approach is used that eliminates the disadvantages of the Kalman snake, which became apparent during the testing.

4.2.2. Redetection Technique

The following approach is used as the main tracking algorithm in this thesis. Bubbles are detected in every frame and their shapes and positions are measured, even if the bubbles are already known to the tracking algorithm.
The bubble positions from the last frame are matched with the bubble position in the current frame to gain the information, which of them belong to the same bubble. This is the direct matching approach. To add reliability a Kalman filter can be employed to predict the bubble position from the last frame before matching. This approach has the advantage of being able to support splitting and merging bubbles. It also allows a more modular design of the software by separating detection and tracking parts. This matching algorithm is introduced in Section 4.2.4. The specifications of the Kalman filter for bubble prediction are shown in the next Section 4.2.3.

If the velocities of individual bubbles are known, dirt or other non-bubble objects can be filtered out, because they do not possess realistic bubble velocities, even if they pass the bubble detection.

4.2.3. Bubble Prediction

The goal of this section is to develop a Kalman filter for the prediction of bubble movements. The idea is similar to that of the Kalman snake outlined in [Sze11], however the method described here is independent of snakes and is also used with other detection algorithms.
The first step is to define a model of general bubble movement. We assume constant upwards bubble motion with vertical and horizontal components. The bubble position in the next frame is assumed to be predictable from the current position of the bubble and its speed.
Let p_{k-1} be the position of the bubble in time step $k-1$ and w_k be noisy influences. v_{k-1} denotes the velocity at this moment. Then the position of the bubbles in time

step 1 can be estimated by:

$$p_k = p_{k-1} + v_{k-1} \cdot 1. + w_k \tag{4.6}$$

As explained in Section 2.3 the Kalman process equation calculates the next state of a system x_k, with model matrix A, previous state x_{k-1}, control input vector u_{k-1} with control transition matrix B and noise w_k as:

$$x_k = A\,x_{k-1} + B\,u_{k-1} + w_k. \tag{4.7}$$

The variable nomenclature for the Kalman filter in this section is identical with the one described in Section 2.3. To use both position and speed of a bubble in the Kalman filter both position and speed are incorporated into the model state as follows.
The state of the Kalman filter at time step k is $x_k \in R^4$. The first two entries of x denote the position of the bubble in the image. p_x is the horizontal position and p_y the vertical position. Both are measured in pixel coordinates. The third and fourth entry denote the velocity v of the bubble. The horizontal component v_x and the vertical component v_y are both measured in bubble movement in pixels per frame.

$$x = \begin{pmatrix} p_x \\ p_y \\ v_x \\ v_y \end{pmatrix} \tag{4.8}$$

The model of predicting the next position based on current position and speed, does not need a control input vector. Hence the matrix B can the disregarded. This simplifies the Kalman process equation to:

$$x_k = A\,x_{k-1} + w_k. \tag{4.9}$$

The assumed constant motion requires that the next horizontal position $p_x(k)$ is be calculated by:

$$p_x(k) = p_x(k-1) + v_x(k-1) \cdot 1 + w(k) \tag{4.10}$$

and the next vertical position $p_y(k)$

$$p_y(k) = p_y(k-1) + v_y(k-1) \cdot 1 + w(k) \tag{4.11}$$

This results in this model matrix A:

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.12}$$

The last two rows resemble the rows of an identity matrix. They ensure Newton's first law, the conservation of momentum. The velocity element entries depend only on the last velocity entries.
For the Kalman filter the measurement noise covariance R must be set. The values R describe the estimated noise in the measurements. Lower values result in a faster response from the Kalman filter. The Kalman filter algorithm incorporates them faster into the modeled process, because it trusts them more.
The process noise w is distributed according to Q, the process noise covariance. The values of Q define the estimated noise in the system itself. It is set according to the expected changes in the system. It defines how much trust is given to previous results from the process. Lower values mean, that the Kalman filter adapts slower to new measurements.

This Kalman filter is designed for predicting the location of single bubbles. For a newly detected bubble, there is no previous information from which to predict a position, therefore for newly detected bubbles a Kalman filter is initialized with its detected position and a rough estimation of the upwards speed of bubbles in this conditions. This value can either be user supplied, or a mean value from past observation of other bubbles can be used. Another option is the mean velocity algorithm as described in Section 4.1.

4.2.4. Bubble Matching

Redetecting the bubble contour positions in every frame, does not directly allow the examination of an individual bubble in different frames. First a matching defining, which bubble position in one frame, together with which bubble position in the next frame belong to same bubble, has to be found. The bubble contours identified in two frames need to be matched to gain the information which contours in which frames belong to the same bubble. This allows the reconstructed of the path of an individual

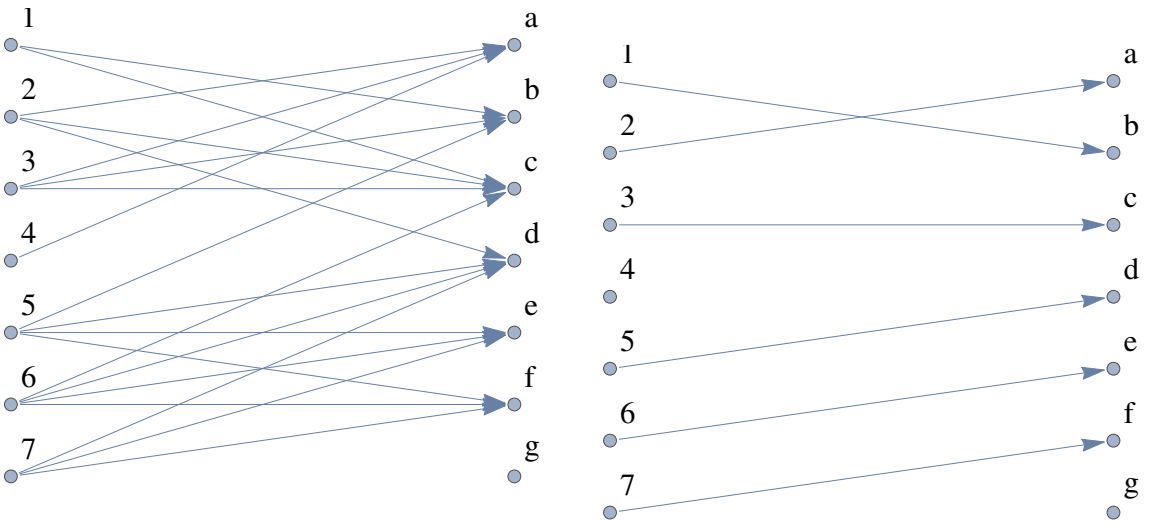
bubble. Before the matching process, the movement of the bubbles, whose positions is detected in the last frame can be predicted with a Kalman filter. In this case, the prediction is used in the following steps, instead of the original bubble position.

To find the best matching, the euclidean norm is measured between the position of every bubble and the predicted or former position of the bubbles detected in the last frame. In the not predicted case, matchings, which suggest non-upwards and non-realistic movement, are discarded. If the prediction is used a maximum distance between prediction and current bubble position is used, to remove matchings which are unrealistic. This can be seen as a projection of the bubble position from the last frame into the current frame and finding the best connection. For an example see Figure 4.6

The realistic assumptions for non-predicted bubbles are necessary, to be able to establish correct matchings in continuous uniform bubble streams. Otherwise, a seemingly correct matching with the doubled movement would be possible. The minimum realistic movement is set to slightly more than half of the typical movement and the maximum realistic movement is set to slightly less than double of the typical movement.

The problem can be represented by a bipartite graph, with the last or predicted positions on the left and the current positions on the right. The matchings that are possible with realistic movement, are shown as edges in Figure 4.5a on the left. Now the best combination of matchings needs to be selected. The solution to the matching problem is shown in Figure 4.5b on the right. It shows a graph allowing all vertices, representing bubbles, to be only part of one edge, which represents a matching. The matchings are selected as to result in the highest possible sum of edge weights, which are calculated from inverse euclidean distances. The edge weights are not shown in Figure 4.5b.
The problem to be solved is known as maximum weighted bipartite matching. By substituting edges with weight 0 for missing edges, it can be transformed into perfect weighted bipartite matching. The Hungarian Algorithm [Kuh55] was the first solution to this problem and is used in this thesis.

The problem can also be solved using flow algorithms [cor09]. For this the inverse weights are used as flow restrictions in an s-t flow. The flow network is created by



(a) Graph of adjacent vertices, meaning bubble positions from two frames.

(b) Maximum weighted matching of 4.5a.

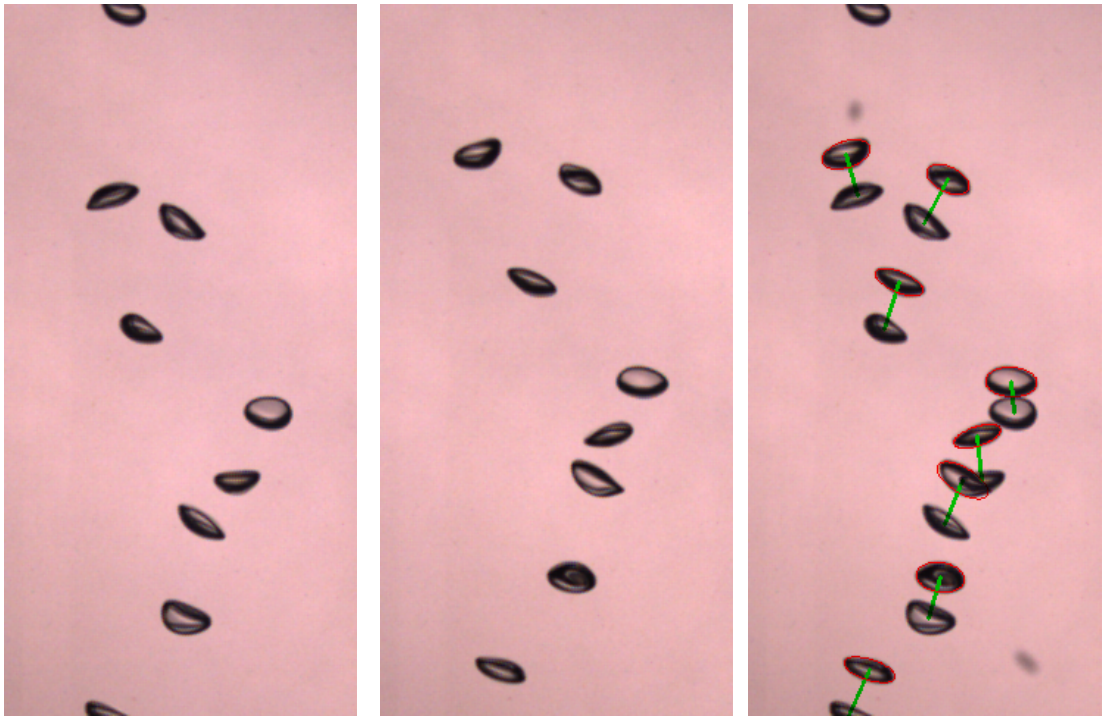
Figure 4.5.: Best matching of a graph. The numbers show the bubble positions in the first frame, the letters the bubble positions in the second. An arrow between them indicates, that the connected number and letter seem to belong to the same bubble. An edge weight relative to the distance of possible matchings is set, but not shown. The maximum weighted matching, the best found combination of bubble contours belonging to the same bubble, is shown in the right image.

connecting the source to one partition, the bubble positions of the last frame and the target to the other partition of bubble positions in the next frame. The maximum flow can be found using the Edmund-Karps algorithm [cor09]. The edges, which are part of the maximum flow, form the maximum weighted matching.

4.3. Volume Measurement

In this section a method for computing the flux of a bubble stream from bubble ellipse approximations and velocities is shown.

The method presented here uses the same assumptions as the method used in [TZSB10]. In the Related Work Section 1.1.1 different methods for measuring the volume are described and reasons for the choice are described. This calculations



(a) Frame 1 (b) Frame 2 (c) Combined

Figure 4.6.: Movement of bubbles in between two frames. The combination shows frame 1 and 2 with bubbles in frame 2 in red and matching bubbles connected in green.

are performed with monocular images, hence no depth information is available. Therefore all bubbles are assumed to be at the same distance from the camera.

Furthermore the bubble ellipse determined in previous bubble detection steps is a projection of the bubble in the viewing plane of the camera, which does not account for the three-dimensional shape of the bubble. However due to the physical forces which influence the bubble shape, in the statistical mean, the three-dimensional bubble shape is rotational symmetric. Thus the depth of a bubble is in the statistical mean of the same size as its horizontal dimension. Therefore the three-dimensional bubble shape can be approximated using an ellipsoid, as shown in Figure 4.7. The volume is initially computed in pixels and then converted using the resolution of the image into metric units.

For the calculation of the volume, the following variable nomenclature is used:

Variable	Description	Dimension
v_y	the vertical movement of a bubble	pixel
fov	the vertical field of view of the camera	pixel
vis	the number of frames a bubble is visible	
a	horizontal axis of a bubble	pixel
b	vertical axis of a bubble	pixel
res	resolution of the image	$\frac{pixel}{mm}$
fps	frames per second	$\frac{1}{s}$
$V_{px}(b)$	volume of bubble b	$pixel^3$
$F_{px}(b)$	flux of bubble b in pixel scale	$\frac{pixel^3}{s}$
$F_{mm}(b)$	flux of bubble b in mm scale	$\frac{mm^3}{s}$
$F_l(b)$	flux of bubble b in liter scale	$\frac{l}{min}$
F_{total}	flux	$\frac{l}{min}$

The volume in pixel units of one bubble b in one frame is given by:

$$V_{px}(b) = \frac{4}{3}\pi\, a\, a\, b. \tag{4.13}$$

This bubble is visible over a number of frames:

$$vis = \frac{v_y}{fov}. \tag{4.14}$$

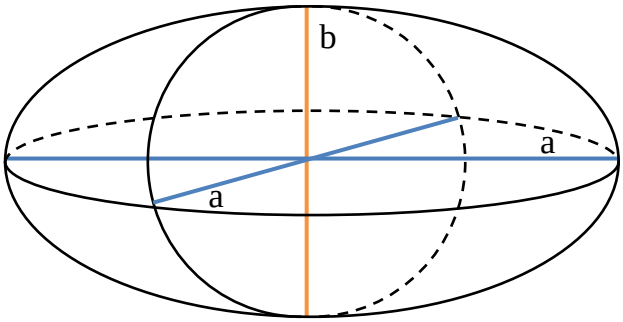


Figure 4.7.: Drawing of a 3D ellipsoid with axes a and b.

The flux in $pixel^3$ per second generated by this bubble b is:

$$F_{px}(b) = vis\ V_{px}(b)\ fps = \frac{v_y}{fov} V_{px}(b)\ fps\,. \tag{4.15}$$

This can be converted to metric units with:

$$F_{mm}(b) = F_{px}(b)\ \frac{1}{res^3}. \tag{4.16}$$

To convert the flux to liter per minute a conversion factor is needed:

$$c = \frac{1}{60 \cdot 10^3}. \tag{4.17}$$

With this factor the flux in mm^3 per second can be converted to the flux in liter per minute:

$$F_l(b) = F_{mm}(b)\, c. \tag{4.18}$$

The total flux is the sum of all bubbles B fluxes and is calculated by:

$$F_{total} = \sum_{i=1}^{|B|} F_l(b_i). \tag{4.19}$$

If a mean speed approach is chosen, no individual bubble speed needs to be

measured. In this case, the sum over all bubble pixel volumes can be calculated directly after 4.13, because v_y is constant.

5. Results

5.1. Experimental Setup

For the acquisition of real bubble sequences and to be able to evaluate the performance of different strategies and algorithms, a laboratory bubble meter has been built. A drawing of the experimental setup is shown in Figure 5.1. This bubble meter consists of a water tank with all sides 40 cm long. The water tank is filled with fresh water.
A perforated hose with an inside diameter of 4 mm and an outside diameter of 6 mm is placed on the bottom of the tank. The nozzle has a diameter of 1 mm and is located on the upper side of the hose. Air is pumped through this hose from an air pump outside of the tank. For a better control of the air flux a flow meter and a flow regulator can be placed between the pump and the perforated hose. Air flow through the perforation results in a stream of rising bubbles, which are observed with a camera.
The camera can be placed either in front of the tank or inside the tank, facing the illumination. As illumination, a 1250 W halogen lamp is chosen. Attached to the lamp body in 40 cm distance is a white diffusor, which is of quadrate shape with a side length of 40 cm. At a distance of 1 m an illuminance of 7000 lux has been measured.
The setup effectively provides a uniform backlight illumination. The choice of backlight illumination and the reasons for a high intensity illumination have been discussed in Section 1.1. An essential advantage of backlight illumination method is that a projection of the shape of a bubble can be observed directly by the dark rim on the contour in the image of a camera. Experiments with other illuminations show a reflection on the bubble, but have a lower contrast between the bubble boundaries and the background.
Typical values for the upwards velocity of bubbles are 25 cm/s [LDLC03]. Thus to acquire good imagery of a bubble, short exposure times and high frame rates are

needed, both require high intensity illumination.
The chosen illumination is bright enough to allow capturing images with high framerate, small integration times, invisible motion blur and a small aperture for a large depth of field. See Figure 5.2 for a photograph of the experimental setup.

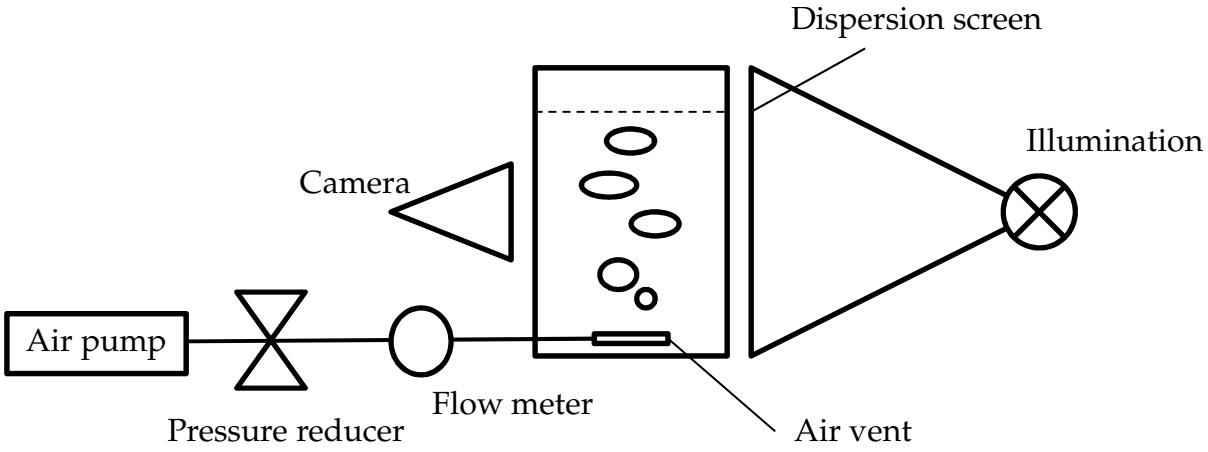


Figure 5.1.: Drawing of the experimental setup. The pressure reducer and the flow meter are optional.

5.1.1. Camera

To acquire images, which allow the evaluation of the bubble size and the tracking of bubbles over a sequence of frames, an appropriate camera has to be chosen. The camera needs a sufficient resolution combined with a high enough frame rate for bubble tracking.

Two kinds of cameras were used to produce the images sequences for this thesis, for pictures see Figure 5.3. The GoPro Hero 3 Black [1]. is a consumer camera, which records on a microSD-card. A picture of the camera is shown in Figure 5.3b. For easier handling it shows a preview of the camera image over a Wi-Fi connected smartphone app. Although it supports an image series mode, high frame rates are only possible in video mode. The camera supports various video modes with different resolutions and speeds. It has a CCD sensor with a native resolution of

[1] see the GoPro Hero 3 Black Edition product manual, available at `http://gopro.com/support/product-manuals-support`

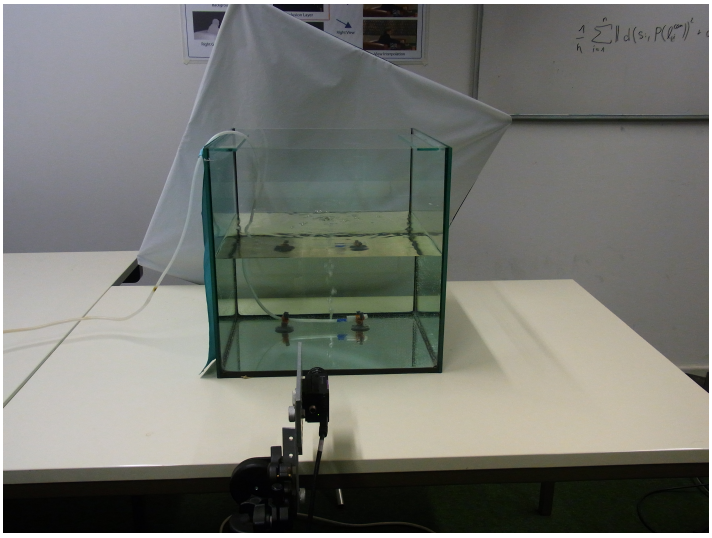


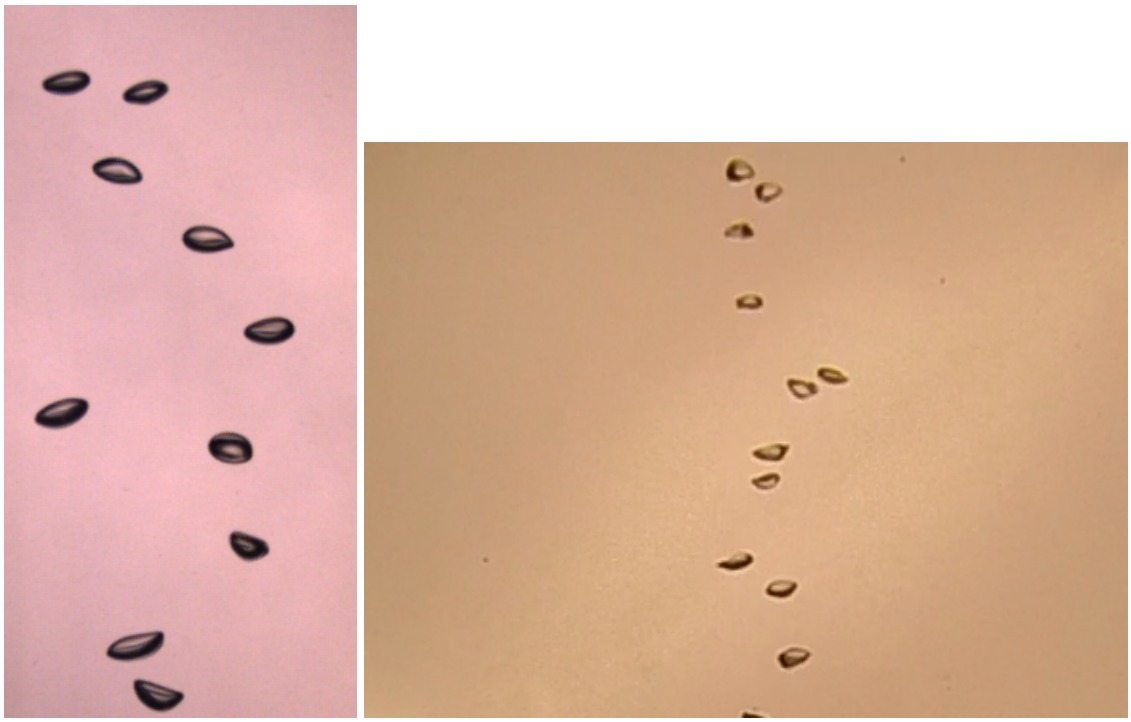
Figure 5.2.: Photograph of the experimental setup.

4000x3000, which captures the images with a rolling shutter. The selected mode results in a resolution of 848x480 and a frame rate of 240 frames per second. It has a field of view of up to 170 degrees depending on the video mode. The video is saved encoded with a h264 compression. Before processing the video file is split into single images, which are cropped to show only the bubbles of interest. The GoPro camera comes with an underwater housing and is located inside the tank within a distance of 30cm from the bubble stream, facing towards the bubbles and the illumination screen.

The second camera used for the experimental setup is the Grasshopper camera depicted in Figure 5.3a. The Grasshopper is an industrial camera with a IEEE1394b interface also known as FireWire produced by Point Grey (see [Gre11]. Its CMOS image sensor has a native resolution of 1600x1200 with a maximum capture speed of 15 frames per second. The camera sensor supports partial readout with higher frame rates with global shutter [Gre11]. The read out speed and with it the frame rate are limited by the height of the selected sensor area. Because of the upwards rising motion of the bubbles, in order to have a field of view which allows the observation of bubbles over more frames and with a high frame rate at the same time, the camera is tilted as shown in Figure 5.2. The Grasshopper camera lacks an underwater housing and is thus located outside of the tank. The resulting images in Figure 5.4a are similar to those found in other research [BDZ+13].

(a) Point Grey Grasshopper 2 (b) GoPro Hero 3 Black Edition

Figure 5.3.: The cameras used in the experimental setup.



(a) Grasshopper image (b) GoPro image

Figure 5.4.: Example images acquired with the Grasshopper and GoPro cameras.

5.2. Ground Truth

Different methods applied on an input image of bubbly flow result in differently detected bubbles. To evaluate the different methods and find out which algorithm provides better detection, an assessment of the quality of the bubble detection provided by the algorithms is needed. In this case, the quality of a bubble detection method is defined by its accuracy in locating bubbles.
There are different ways of assessing the quality these algorithms. For small data sets, it is possible to make a visual comparison of input image with drawn in bubbles from different detections and decide which one is more accurate. However this approach is inaccurate for small differences and impractical for larger image series. Sufficiently large image series are necessary to produce a reliable result, as results are distributed statistically.

The direct way to acquire information about the quality of the detection is to compare it with ground truth data. Ground truth (GT) data is manually obtained by marking the bubbles in the image, in such a way that the markings can be easily extracted. Figure 5.5 shows a bubble marked in green as the ground truth in an input image.

To evaluate the quality of the bubble detection, different areas in an image can be separated:

1. The area where detection and ground truth data match. This can be seen as the true positive values.

2. The area where a bubble is detected, but there is no marking in the ground truth data. This can be seen as the false positive values of a detection.

3. The area is marked in the ground truth data, but no bubble is detected. This can be seen as the false negative values.

As a measure of quality the comparison of the detection area, ground truth data and the marked area is introduced. This differs from [TZSB10] which measures the difference in size from manually measured ellipses. Let D denote the set of points describing the detection and GT the set of points of the ground truth as shown in

Figure 5.6, then the positive matching ratio p is attained by:

$$p = \frac{2\,|D \cap GT|}{|D| + |GT|}. \tag{5.1}$$

p attains values between 0 and 1. Higher values show a higher matching rate between ground truth and detection, which means a higher quality of detection. Using both detection and ground truth count in the denominator makes this measure sensitive to all areas 1,2,3 defined above. This is an advantage to the measure as introduced in [BMKD12], which measures the absolute difference between ground truth and detection divided by the size of ground truth.
The error rate e is defined by:

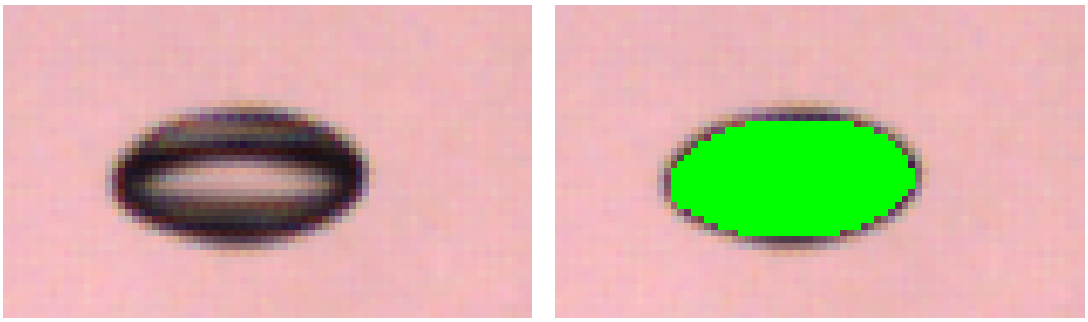
$$e = \frac{|GT \setminus D| + |D \setminus GT|}{|D| + |GT|}. \tag{5.2}$$

It can be shown that e is the opposite measure of p and that

$$p = 1 - e. \tag{5.3}$$

The positive measure p will be used as the primary measure of detection quality in this thesis, it is therefore also called detection rate. This allows direct comparison between the results of different algorithms. Because this does not show the source of potential misdetection, visual inspection of the original image with drawn in detection is used as secondary test.

The norm introduced in this section is based on sets of points, which can easily be applied to sets of pixels. The ground truth data is drawn pixel based, as shown in Figure 5.5. An ellipse is the results of the CMA-ES method and the Canny edge method. Moreover it is used in the volume measurement. To allow both exchangeability through modularity in software and ensure the comparability between the detection rate results from CMA-ES methods and the snake method, the ellipse is chosen as the input for the ground truth detection presented here. Therefore, as described in Chapter 3, an ellipse is fitted into the result of the snake algorithm, with the same method as in the Canny edge method.



(a) Cropped original image (b) Ground truth marked in green

Figure 5.5.: Original image and corresponding ground truth. Both pictures are scaled up by factor 4. This image was acquired using a Grasshopper camera.

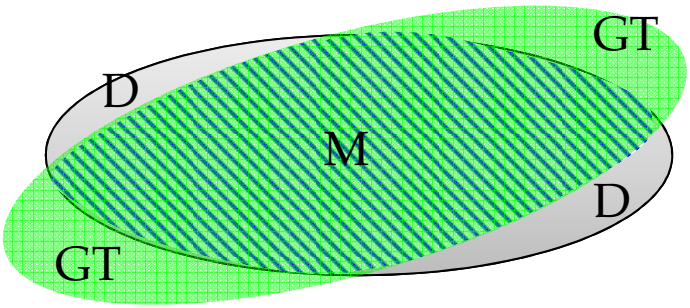


Figure 5.6.: Drawing of ground truth and detection. The detection result D is shown in gray and the ground truth GT in green. The matching area $M = GT \cap D$ is highlighted in blue stripes.

5.3. Bubble Detection

5.3.1. Snake parameters

The classic snake algorithm as described in Section 2.1 has three parameters α, β and γ. They set the weighting for the different terms of the snake energy functions as described in Section 3.3. α weights the continuity term, β the curvature term and γ the gradient term. The shape of the snake and the quality of the detection depends on this parameters. In the original paper about the snake algorithm [KWT88], the choice of parameters is left to the user. A particular set of parameters that worked is noted, but not how it was derived. In this thesis different combinations of α, β and γ are tested against ground truth data.

For this experiment an series is recorded using the Grasshopper camera. The flux is set to a level, where the bubbles are clearly visible. The physical setup is described in Section 5.1. The frame rate is set to 100 frames per second. The images are recorded with 146 pixel width and 1120 pixel height and are cropped to the size of 146 pixel width and 425 pixel height. The image resolution is 3.8 pixel per mm, measured with a ruler at the same distance as the bubbles in a test image. The bayer pattern of the camera is converted to 8 bit gray scale values. The first frame of the image series is depicted in Figure 5.4.

In this experiment, bubbles are detected in each image. First a coarse localization of the bubble is found and around this localization the snakes are initialized. For more detailed information about these algorithms see Chapter 3.
The level of correspondences between ground truth and snake detection is calculated as described in Section 5.2 and Equation 5.1. The higher the value, the better the detection.

The results are depicted in Figure 5.8. For different values of α, the β and γ parameters, the level of correspondences between ground truth and snake detection averaged over the image series is shown in respective contour plots. The parameters α, β and γ weight the continuity, curvature and gradient terms of the snake energy function. Brighter colors show better detection. The bright areas dominate the graphics, showing good results. A restriction, which becomes apparent in Figure 5.8 is, that to ensure good results, α should not be near zero, otherwise more care has to

be taken about β and γ. Furthermore, the combination of high values of γ with low values of β does not provide the best results, as shown in the images for $\alpha = 0.2$ and $\alpha = 0.4$. Aside from this, the results show, that a wide range of parameters deliver good results and that the choice of parameters for the snake algorithm is not critical.

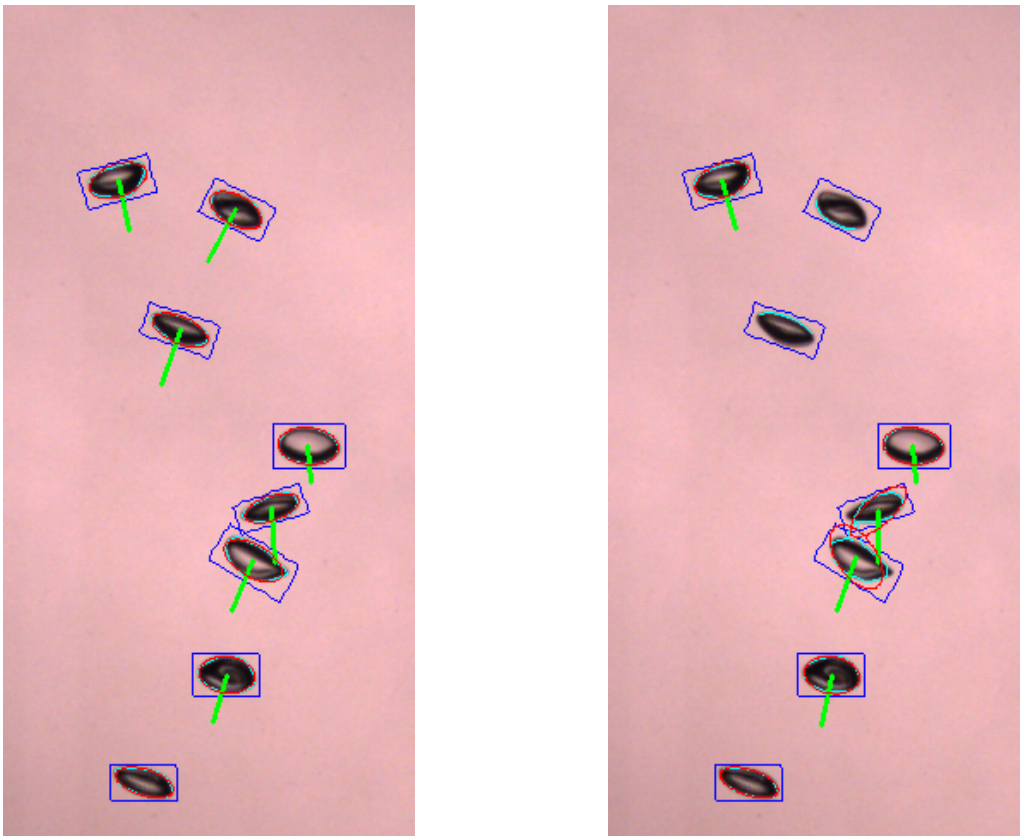
The results have been confirmed by visually comparing the snake contour with the bubble images for certain parameter combinations. As an example the bubble detections with different parameters on a Grasshopper image are shown in Figure 5.7. On the left image a good bubble detection with parameters $\alpha = 0.4$, $\beta = 0.7$ and $\gamma = 0.15$ is shown, while on the right image a significantly higher γ value of 0.8 was used, which means that the image gradient is weighted much higher. The higher γ values lead to different misdetections, such as collapsing snakes (see bubbles 2 and 3, counted from the top) and snakes which converge onto the wrong bubble (see bubbles 5 and 6, counted from top). The first happens, if the curvature is not weighted high enough, relative to the image gradient and the second, if the image gradient is weighted too high, compared to the continuity term. This confirms the results from Figure 5.8. This figure shows that the parameters of the left image result in a high detection rate and thus in a bright area, while the darker colors for the parameters of the right image mark lower detection rates.

To be able to generalize these results, this evaluation is also performed with images captured by the GoPro camera. The original GoPro images have 840 pixel width and 484 pixel height. The image resolution is 4 pixel per mm, measured with a ruler at the same distance as the bubbles in a test image. The results are shown in Figure 5.9. The first frame is depicted in Figure 5.4.

The light areas showing good results are even more prominent, than in the results from the Grasshopper images series. A visible restriction is that very low values of γ are to be avoided, as well as the combination of high β value in combination with low γ values in the presence of a high α value, see Figure 5.9f. However these restrictions are minor compared to the large range of parameters with good results. The GoPro images show more even results with fewer dark spots inside bright areas. This is caused by the higher count of small bubbles in the GoPro camera, which even out unfavorable results and completely undetected. Other factors are that the detection rate, even with outher non-snake methods, for the GoPro image sequence is generally a higher (see next section) and the smaller variation in bubble shapes

and sizes occurring in the GoPro images as visible in Figure 5.4.

These results confirm the results attained from evaluation of the Grasshopper image series. The snake algorithm on these images provides good results for a large range of different combinations of parameters. Furthermore, it is possible to find parameters, which result in a good bubble detection in both image sequences, for example $\alpha = 0.4$, $\beta = 0.7$ and $\gamma = 0.15$.



(a) Parameters: $\alpha = 0.4$, $\beta = 0.7$ and $\gamma = 0.15$. (b) Parameters: $\alpha = 0.4$, $\beta = 0.7$ and $\gamma = 0.8$.

Figure 5.7.: Bubble detection with the classic snake algorithm and different parameters on Grasshopper images. The parameters α, β and γ weight the continuity, curvature and gradient terms of the snake energy function. The coarse localization, which is the initialization of the snake is depicted in blue, the final snake in cyan and the ellipse in red. The green line shows the movement of the bubble from last frame.

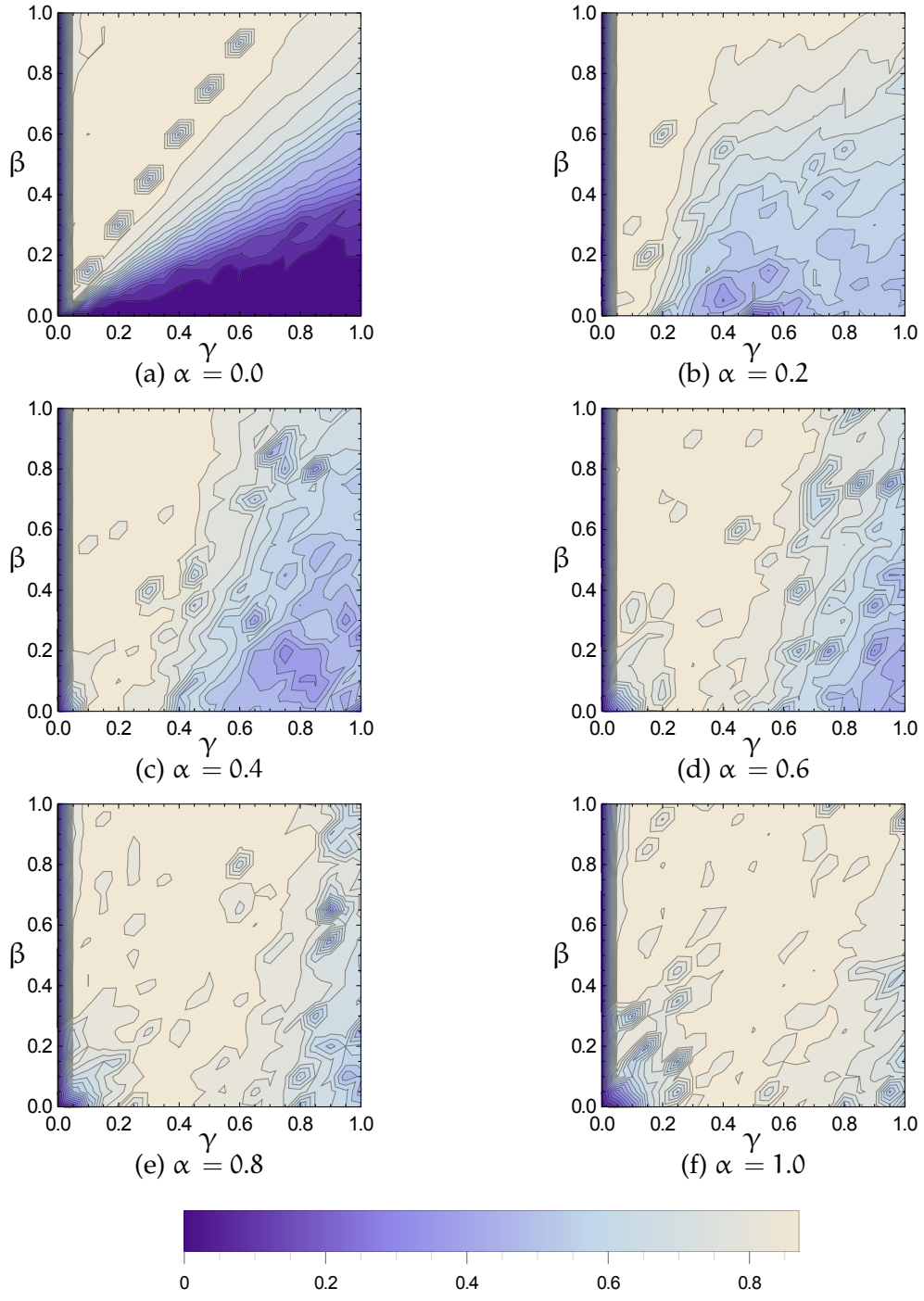


Figure 5.8.: Contour plots of detection rates p with different parameters for the classic snake algorithm on Grasshopper images.

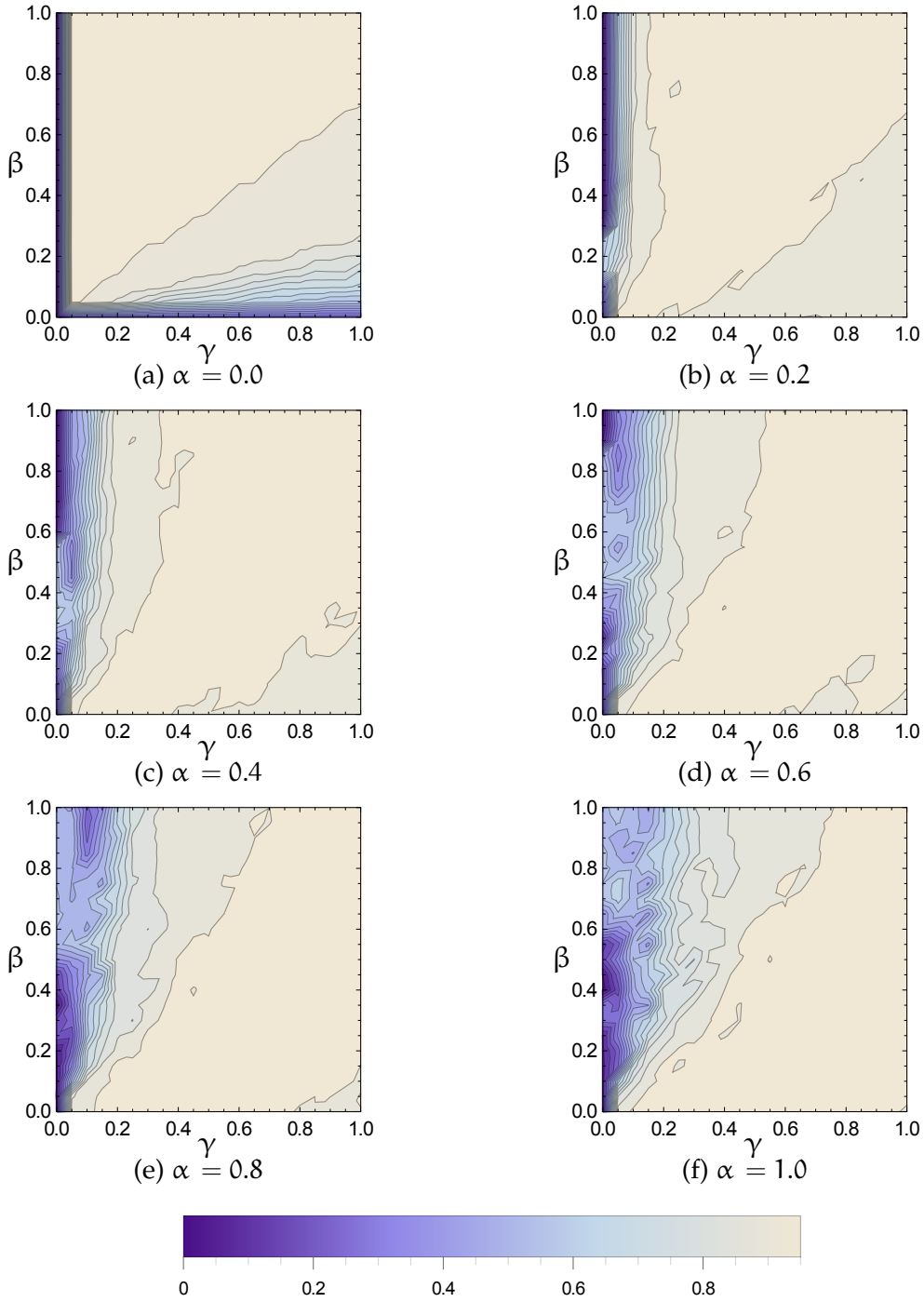


Figure 5.9.: Contour plots of detection rates p with different parameters for the classic snake algorithm on GoPro images.

5.3.2. Methods

In this section the different methods introduced in Chapter 3 are compared with each other and the results are analyzed.
Two image series were captured with the same settings as in Section 5.3.1. One is acquired with the Grasshopper camera and the other with the GoPro camera. For a visual impression the first frame of each image series is depicted in Figure 5.4. The following table gives an overview over the different methods used:

Method	Short Description	Reference
Snake	classic snake algorithm	Section 3.3, Eq. 2.2
Refinement snake	snake algorithm with coarse and fine stage	Section 3.3.3
Gradient direction snake	snake factoring in the gradient direction	Section 3.3.2, Eq. 3.3
CMA-ES edge gradient	the mean image gradient on the edge of an ellipse	Section 3.4, Eq. 3.6
CMA-ES edge intensity	the mean intensity on the edge of an ellipse	Section 3.4
CMA-ES difference of means	the difference between the intensity means inside and outside of an ellipse	Section 3.4, Eq. 3.11
CMA-ES average difference from mean	the average difference from the mean inside or outside intensity per pixel	Section 3.4, Eq. 3.12
CMA-ES combined	sum of the two function directly above	Section 3.4, Eq. 3.13
Canny edge	Canny edge method	Section 3.1

For the snake algorithm the parameters $\alpha = 0.5$, $\beta = 0.3$ and $\gamma = 0.2$ and for the gradient direction snake the parameters $\alpha = 0.13$, $\beta = 0.7$ and $\gamma = 0.2$ are used. The results for the Grasshopper image series are shown in Figure 5.10 and the results for the GoPro image series in Figure 5.11.

The low detection rate of the canny edge method in the Grasshopper sequence is a result of misdetection of the inner contours of the bubbles. They can be detected by the canny edge detector as separate contours and sometimes be a part of the bubbles outside contour. An example for this type of detection is shown in Figure 5.12b. For the other methods the influence of this kind of error is limited due to the bounding box prior in the coarse localization step.

The snake algorithm with and without gradient direction showed the best results in the Grasshopper image sequence and very good results in the GoPro image sequence. In both image sequence the difference in the detection rate between the snake method with and without gradient direction is negligible. Because of the overall high detection quality of these methods and the lack of non-detections or completely false detections of bubbles, the improvements in detection quality that the gradient direction snake possesses do not become readily apparent. The improvements in detection quality that the gradient direction snake shows can however be seen in the examples in Figure 5.13 where the camera images are of low quality. The shapes of bubbles 1 and 2 counted from the top of the bubbles in Figure 5.13 are falsely detected by the classic snake algorithm, because the snakes shrink to much, due to all image gradients being weighted equally. In contrast, the gradient direction snake on the other hand detects the shape of these bubbles correctly.
The refinement snake shows a slightly lower performance. It seems that the low resolution snakes are not as stable as the higher resolution snakes. They do not locate reliably the outer contour of the bubble, which is a local minimum for the snake energy function, but collapse instead.

An example for a snake based detection is depicted in Figure 5.12c. It also shows that even for overlapping bubbles, the contour is correctly detected, although not split into two. Therefore, if a bubble stream with many overlapping bubbles is measured, the snake algorithm still shows good results, although with an overestimation in the bubble size. The overall good performance of the snake methods shows that they are a good choice for bubble segmentation.

From the edge based CMA-ES methods the gradient approach shows better results, than the edge based image intensity method. The CMA-ES edge intensity approach minimizes the intensity per ellipse edge point. The lower detection rate in both image sequences of the CMA-ES edge intensity method is caused by a constant underesti-

mation of the bubble size, as shown in Figure 5.12a.
The explanation for this behavior is that the ellipse through the darkest areas of bubble is not located on the outside of the dark rim of the bubble, where it could intersect with areas of higher intensity, but placed directly on the dark rim inside of the bubble.
The CMA-ES fitness function based on the bubble edge gradient shows better results. It only evaluates the edge of a bubble and is therefore generally strongly dependent on the ellipse form of the bubbles. Because the bubbles in both image sequences can be approximated very well with ellipses, this does not to influence the results negatively.

The results of the Grasshopper image sequence show that with the CMA-ES methods, the fitness functions based on the difference of the mean intensity f_s and on the combined approach f_c have a better detection rate than the fitness function based on the difference of the mean f_μ. On the GoPro image sequence the opposite is the case. The reason for the lower performance of the difference from mean based methods is that in the GoPro image sequence, only the rim of the bubbles has a prominent low intensity, which can be seen in Figure 5.4b on an example image. This rim is sufficient to provide acceptable results. The inside mean is formed from the complete inner ellipse area between the brighter inner area and the rim. However the difference from this mean to all inner points is not as uniform as the rim and the inner area has a higher variation in intensities. This type of fitness function is intended to separate two areas with different mean intensities.
This is the case in the Grasshopper image series (for an image see Figure 5.4a) and the reason, why the methods based on fitness functions, which use the difference from the mean value as a criterion, work so well on this image series and provide the best results of all methods. It follows that the difference of means approach is the most versatile approach from the area based CMA-ES based methods, while the combined approach should be preferred if the bubbles have a overall lower intensity than the background.

5.3.3. Bubble Size Distribution

In order to test, whether the bubble size follows a normal distribution, an image sequence of 800 images was recorded with the Grasshopper camera. The image series

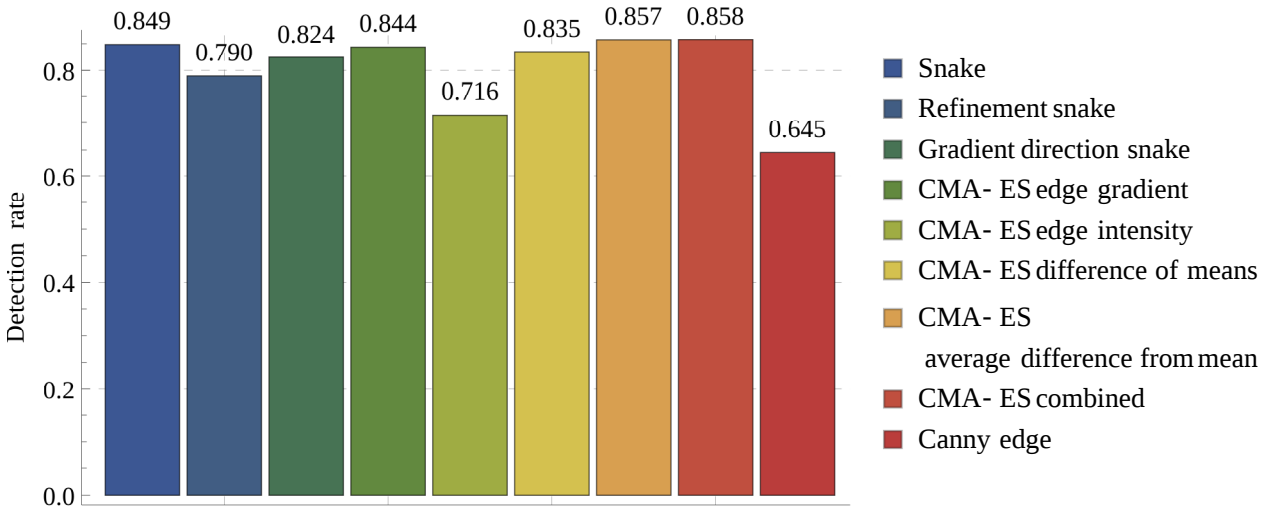


Figure 5.10.: Bar chart of detection rates with different methods on Grasshopper images.

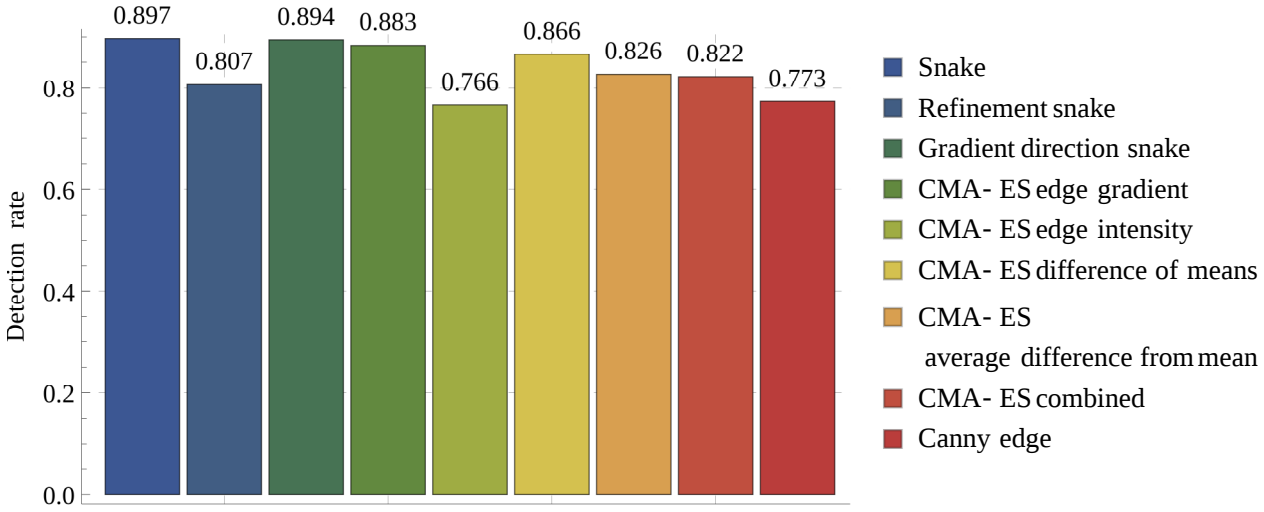
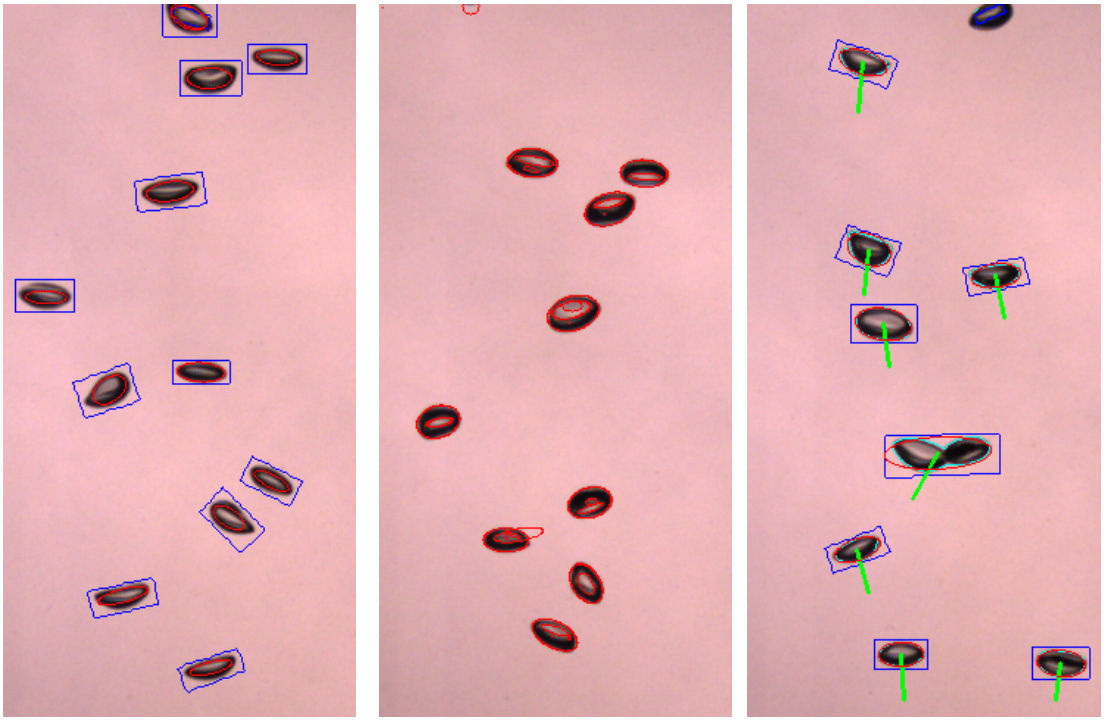
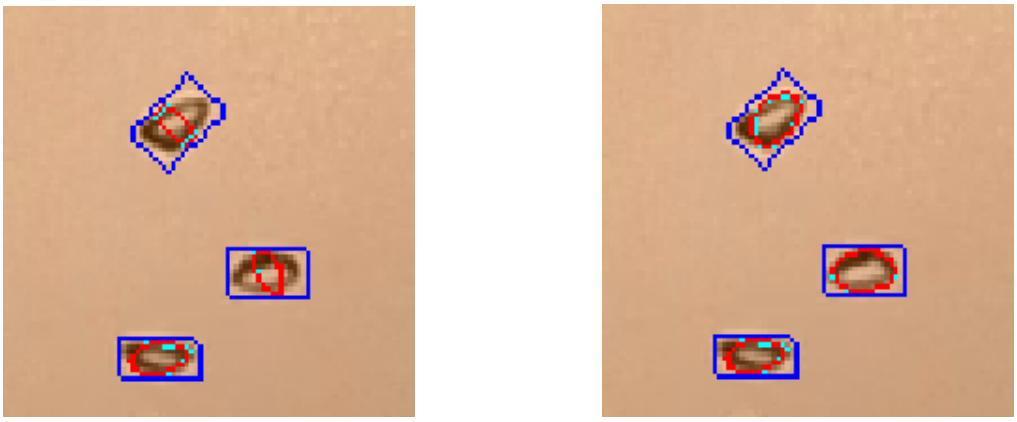


Figure 5.11.: Bar chart of detection rates with different methods on GoPro images.



(a) CMA-ES edge value (b) Canny edge method (c) Classic snake algorithm

Figure 5.12.: Images of the bubble detection results with different detection algorithms. In image (a) a detection with the CMA-ES based method, which weights the image intensity values of the edge of the ellipse is shown. The image (b) shows the results from the Canny edge method and the image (c) the detection using the classic snake algorithm without gradient direction. The coarse localization is shown in blue, while the snakes are shown in cyan and the ellipses are shown in red.



(a) The classic snake algorithm.

(b) The gradient direction snake algorithm.

Figure 5.13.: This figure shows the improvements in detection quality that the gradient direction snake provides over the classic snake algorithm. Depicted are cropped detection results of a bubble detection using the classic snake algorithm and the gradient direction snake algorithm on the same image with the same parameters: $\alpha = 0.4$, $\beta = 0.7$ and $\gamma = 0.15$. This image is especially selected, because of lower image quality. The coarse localization is shown in blue, while the snakes are shown in cyan and the ellipses are shown in red.

has a frame rate of 100 frames per second. All values in this section are in pixel unless otherwise specified.
The image series is evaluated using the CMA-ES average difference method and the detected contours are combined to tracks of individual bubbles with the individual bubble matching approach. The average size over all frames where it is detected is calculated for every detected bubble.
The data is collected in a histogram shown in Figure 5.14. The data is distributed with a mean of $\mu = 13.61$ and standard deviation of $\sigma = 2.44$.
A χ^2 hypothesis test is performed to test for a normal distribution. In this case the bubble size distribution is normal distribution with $\mu = 13.61$ and $\sigma = 2.44$. For more details about the χ^2 test see [BS84, p.689].

The null hypothesis is that the data is normally distributed with $\mu = 13.61$ and $\sigma = 2.44$. It is tested on a 0.1% significance level. The result is a p-value of 0.62, which means that the null hypothesis is not rejected at the standard 5% significance level.
Figure 5.15 shows the cumulative frequency histogram together with the cumulative probability density function of the normal distribution. The good fit confirms the results from the hypothesis test. The fit is very close, which proves that the bubble sizes in a bubble stream are normally distributed.

5.4. Bubble Tracking

The tracking of bubbles means establishing the matching relationship between two contours of the bubble positions in two frames. Two methods are introduced in Chapter 4, the mean velocity approach and individual bubble matching with Kalman prediction. This section presents results from these methods.

Tracking results for the Grasshopper camera is shown with a green connection together with the two original frames in Figure 4.6. In Figure 5.16 the tracking of bubbles in another frame is shown, together with the tracking of a frame, captured with the GoPro camera. Visual inspection of tracking images like these confirms that both methods for the individual bubble tracking and the bubble detection work very well.

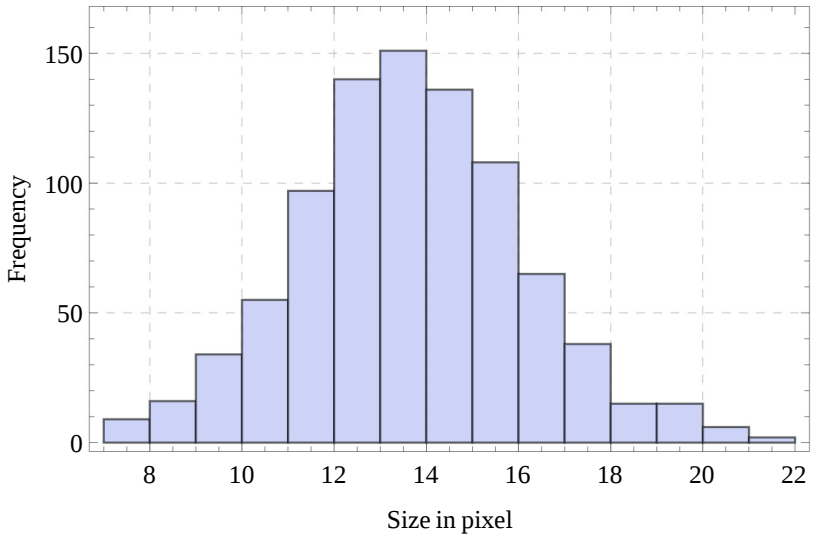


Figure 5.14.: Histogram of the bubble size distribution of a Grasshopper captured image series with 800 images.

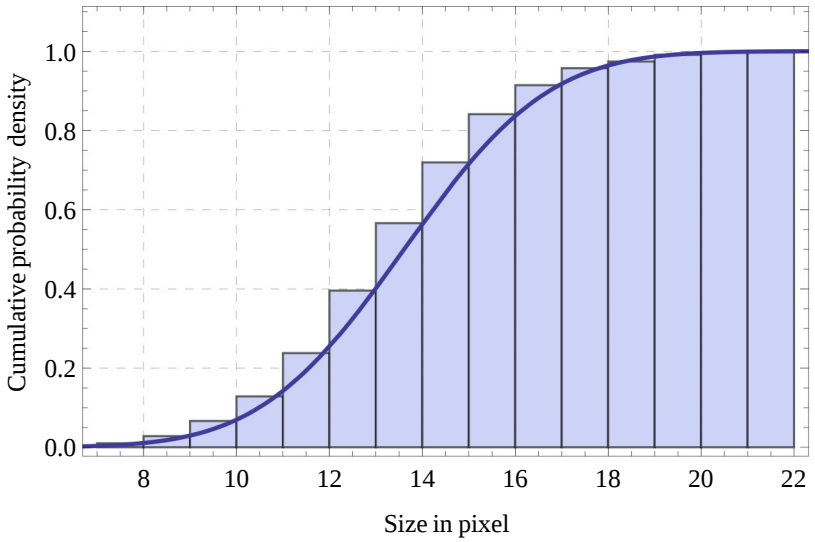
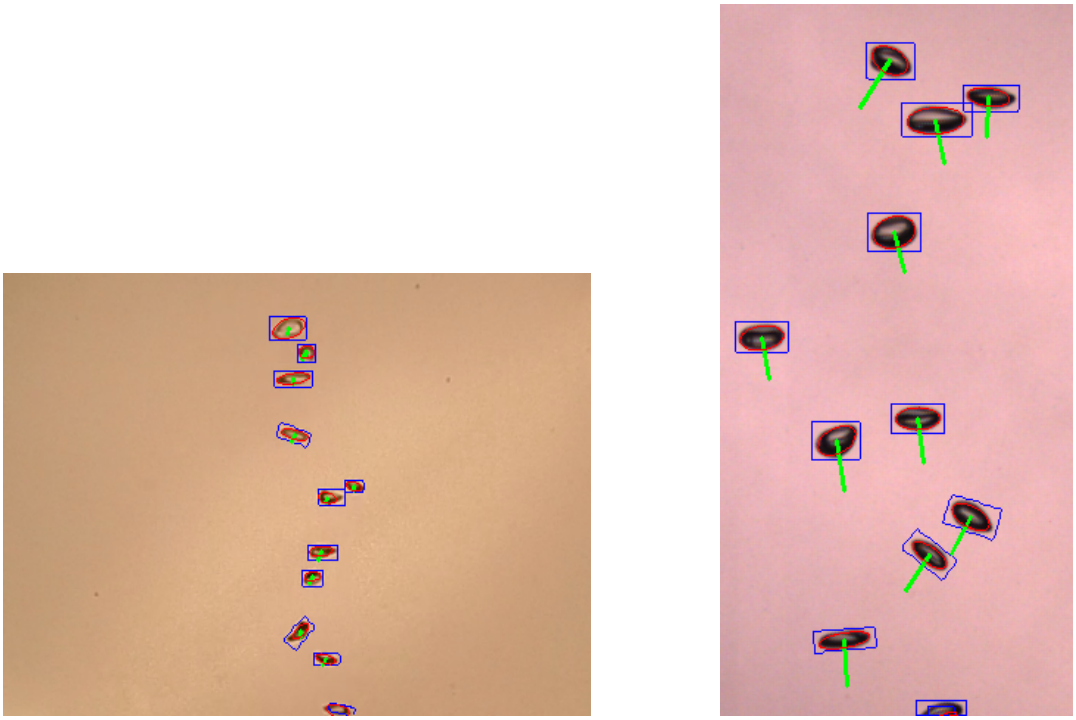


Figure 5.15.: CDF (cumulative probability density function) of the fitted normal distribution with $\mu = 13.61$ and $\sigma = 2.44$. The normalized cumulative size frequency distribution is shown in blue bars.



(a) Image sequence captured with GoPro camera.

(b) Image sequence captured with Grasshopper camera.

Figure 5.16.: The red ellipse depicts the detected ellipse, while the green line shows the movement of this bubble from the last frame.

To compare the difference between the mean velocity result and individual bubble tracking, the average velocity in an image series of 10 frames from the GoPro camera is calculated with both methods. For the mean velocity approach this result is calculated by taking the average of the individual frame velocities, which results in a mean velocity of 4.329. For individual tracking the mean is calculated by averaging the mean speeds of the individual bubbles, weighted with their occurrence, resulting in 4.003. The mean velocity result is 8% larger than the individual result.

For the mean velocity approach remains the main question, whether its critical assumption is valid. This assumption is that the velocities of individual bubbles in a bubble stream are distributed in such a way that creating a mean velocity is feasible and valid. The normal distribution is a candidate.

The bubble velocity is distributed with mean $\mu = 13.23$ and standard deviation $\sigma = 1.29$. A histogram of the distribution is shown in Figure 5.18. The χ^2 hypothesis test is performed to test for a normal distribution. The null hypothesis is that the bubble velocity is distributed according to a normal distribution with the same parameters μ and σ.
The p-value of the χ^2 test is 0.252. Hence the null hypothesis is not rejected at a 5% significance level. The low difference of the cumulative normal distribution from the cumulative speed distribution in Figure 5.19 shows, that the velocity distribution in a bubble stream is also normally distributed.

The previous section showed that the bubble size is normally distributed and that the upwards bubble velocity is also normally distributed. The low covariance value of 0.130 between size and velocity data strongly suggests that the two parameters are independent. This can be confirmed by observing the spread of values in Figure 5.18. It follows, that in a bubble stream the bubble velocity and bubble size are independent and normally distributed. The normal distribution implies that a common mean value exists. This validates the mean velocity approach for bubble streams as produced in the experimental setup.

Please note that caution must be applied if one tries to generalize these results to free water bubbles. These results were obtained in a small water tank. It is possible that in the small tank the stream of rising bubbles forms a convection cell. Due to friction the water surrounding the bubble could also be moving upwards averaging the velocity of the bubbles.

5.5. Volume Measurement

The goal of this experiment is to test the quality of the quantification of the air flux, possible with the components developed in this thesis.

Because a calibrated flow meter was not available, the reference values are obtained by measuring the time needed to fill a calibrated measuring cup. This method integrates the flux over the measuring time directly. The timings are then converted to liters per minute. Otherwise the experimental setup remains unchanged.

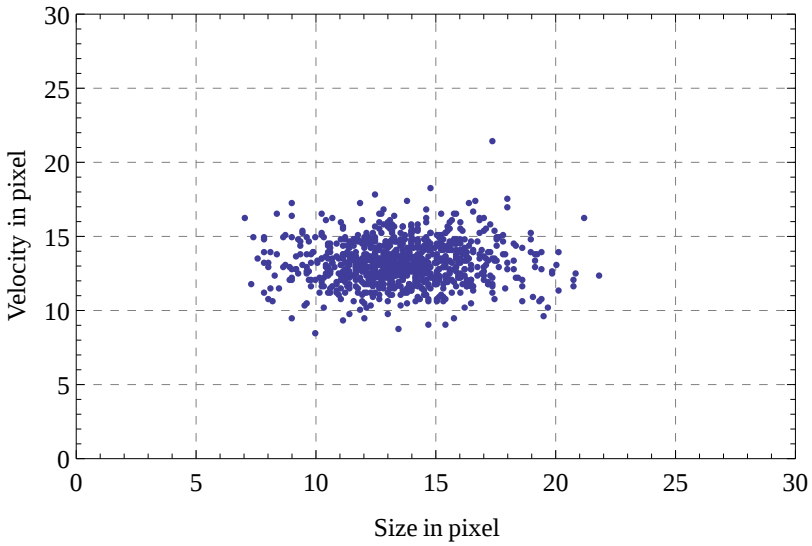


Figure 5.17.: Size and velocity distribution in pixel.

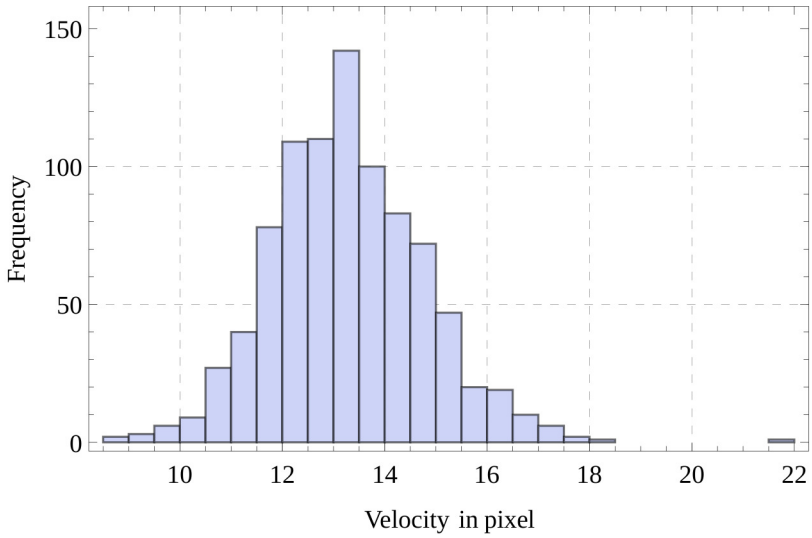


Figure 5.18.: Histogram of the velocity distribution.

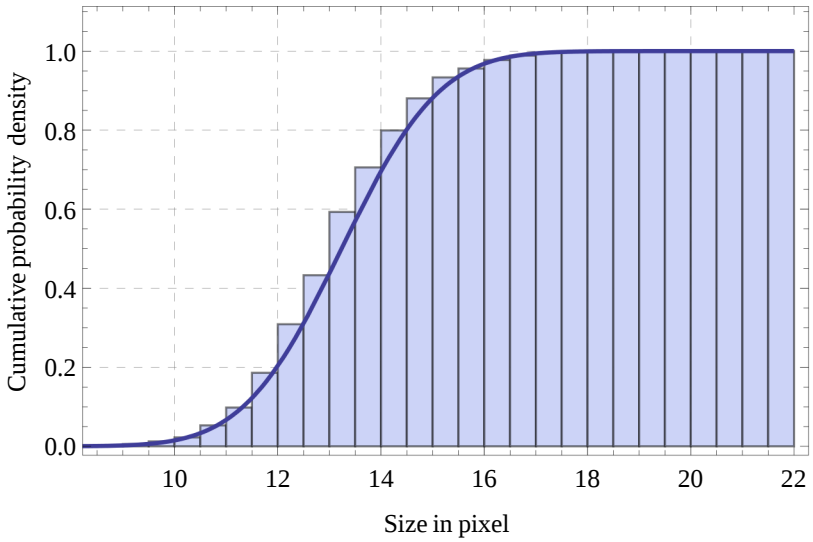


Figure 5.19.: CDF (cumulative probability density function) of the normalized velocity frequency distribution data, assuming a normal distribution with $\mu = 13.23$ and $\sigma = 1.49$.

For this experiment, the flux is set to three different levels of flux. For each level the flux is measured. Directly afterwards, an image series is acquired using the Grasshopper 2 camera and a frame rate of 105 frames per second. Each image series consists of 300 frames. The position of the bubble stream and the camera are fixed. The camera is calibrated at the depth of the bubble stream, using an in place scale and measuring the resolution in pixel per mm. As a bubble detection algorithm, the CMA-ES average difference method is chosen. The detected bubbles are tracked, their velocity is measured with the mean velocity approach and their volume is measured according to Section 4.3. This tracking method velocity is valid, as confirmed in Section 5.4.

The different values measured and calculated from the results of the bubble detection, tracking, and volume calculation are compared in Figure 5.20.

The results of the measurement and the calculation differ only slightly. The larger over estimation occurring at larger flux, compared to the medium or small flux, could be explained with more overlapping bubbles, which are detected as larger bubbles and therefore lead to an over estimation.

The results confirm the correctness of the detection and tracking methods used, as well as the volume measurement formulas.

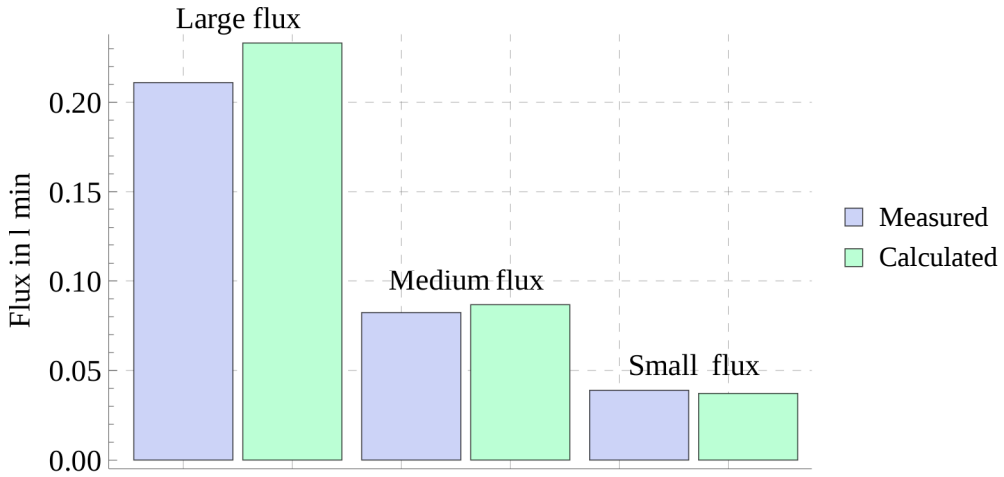


Figure 5.20.: Comparison between measured and calculated flux in liter per minute.

6. Summary

In this thesis different methods for bubble detection based on the snake algorithm and the CMA-ES optimization algorithm were developed and implemented. The bubble detection is designed as a two-step approach with an edge sensitive coarse localization step and a snake or CMA-ES based fine localization and shape fitting step.

The accuracy of different methods has been evaluated with a ground truth based approach. The results show, that the snake algorithm shows good results and that finding working parameters for it is not an issue.
The optimal choice of parameters for the snake algorithm is not specified in the known literature. This topic has been extensively studied on image sequences acquired with two different cameras. The result is that the snake algorithm is robust and well suited for the detection of bubbles, as the choice of parameters is not critical. The result is of practical importance for the design of a bubble meter.

A novel enhancement of the snake algorithm by modifying the snake to factor in the direction of the image gradient, called the "gradient direction snake" is introduced in this thesis. This modified snake algorithm provides a noticeable improvement in detection quality.

Moreover a novel refinement technique for accelerating the snake algorithm has been devised, though at a cost of detection quality. Future work to diminish or rescind this disadvantage is advised.
Different fitness functions for the CMA-ES based ellipse fitting step were devised and implemented. The results varied depending on the camera image properties. If the bubbles are dark, the functions evaluating the ellipse area show the best performance, otherwise the functions, which use only the rim of the ellipse, perform better. The best performance in both situations is shown by the fitness function, which maximizes the difference between the mean intensities of inside and outside the ellipse and the fitness function, which maximizes the gradient values on the edge of the ellipse. Both

the snake based and the CMA-ES based detection methods provided better results, than a baseline canny edge bubble detector established by previous work.

To enable the analysis above and for further research a monocular backlight illuminated experimental setup has been designed, allowing a bubble measuring system with the following steps. Bubble detection, bubble tracking and flux volume calculation.
In this thesis two different bubble tracking methods have been studied. Firstly the mean velocity method, which provides the easy access to the mean velocity of all observed bubbles and the individual bubble tracking with Kalman prediction, which accumulates the size and secondly velocity of individual bubbles in a data structure for further processing. This data can be used for further research at the users discretion. Results show that size and velocity of individual bubbles inside a bubble stream are normally distributed and statistically independent. This confirms the principal feasibility of the mean velocity approach. The developed detection and tracking methods together with volume determination formulas derived in this thesis allow measuring of the gas flux in a bubble stream in liter per minute. The accuracy of the flux volume measurements shown in the results section confirms the validity of the prior bubble detection and tracking steps.

Even though the research of this thesis into bubble detection and tracking already shows good results, the methods and the bubble meter can be further improved in different ways. Several future work directions are listed as follows.

Multi-camera Rig With a stereo or multi-camera rig the position of the bubbles can be triangulated. Furthermore the shape of the bubbles can be reconstruction more accurately using more perspectives. This would allow the extension of the currently narrow fixed depth of the bubble stream and a better shape reconstruction, together with better volume measurement.

Calibration Currently the images are calibrated by measuring their resolution in pixel per mm. A special under water calibration [JSK12], which accounts for the refraction on water to glass interfaces, seems to be a good way to provide more accurate results, especially if a multi-camera rig is used and depth information is available.

Optical Flow Sparse or dense optical flow algorithms can be used to support the

detection of bubble contours and to directly estimate their movement.

Better Approximation In this thesis the bubbles are approximated with ellipses. Other approximations of higher order seem to be a promising focus of future work especially for more irregular bubbles in more turbulent bubble streams.

Optimized Optics for GoPro The GoPro cameras have fixed focus optics, requiring an underwater object distance of approximately 30cm for reasonable sharp images. Focusable optics would avoid this restriction.

Level Sets Snakes and and the ellipses for CMA-ES optimization need to be initialized with the coarse location of the bubbles. Using level-set algorithms, which define contours implicitly with the level-sets (sets of points resulting in the same value) of an underlying function, this step can be eliminated.

Intensity Patterns Inside of Bubbles The bubbles acquired with the Grasshopper camera (see Figure 5.4a) show a difference in intensity inside the bubbles. While there is mostly a bright spot inside the bubble, the exact intensity pattern is individual per bubble. This intensity profiles seems to be caused by a differences in curvature in the individual bubble shapes. This effect could be used to allow a more accurate shape reconstruction.

Strobe Illumination With the short exposure times necessary to prevent motion blur, much light is wasted in between frames. Using a synchronized strobe the total illumination could be reduced.

Whole Frame Detection and Tracking Eliminate need for cropping, detect bubbles in complete frame, remove non-bubble object in tracking phase, learn about area where bubble occur.

Extended Bubble Prediction An occlusion occurs, if two distinct bubbles differs only in their distance from the camera and are mapped onto the same position in the camera image. Certain bubble occlusions are predictable from the movement of bubble prior to the occlusion. An advanced prediction of the bubble movement could handle this case and provide more accurate results, especially in turbulent or dense bubble streams.

Overlapping Bubbles An introduction to different techniques used for overlapping bubbles is presented in [MP03]. The main techniques introduced are the curvature profile method and the convex perimeter method.

In [TZSB10] several overlapping bubbles are encompassed in one ellipse. This results in an over-estimation of the flux, which is compensated by introducing a constant factor.
Detecting overlapping bubbles and using algorithms to discern individual overlapping bubbles in bubble clusters should increase the accuracy of the bubble meter.

A. Physics of Bubbles

The following physics laws can be applied to bubbles in a laminar flux.

Stokes Law If a bubble is approximated as a solid sphere, which is propelled upwards by the buoyant force of the density differences between air and liquid, Stokes law can be applied to calculate its drag force. Stokes law (see [Czi91]) states that the drag force for a velocity U of a solid sphere of radius R dragged through a Newtonian fluid with viscosity η is

$$F_w = 6\pi\eta RU. \tag{A.1}$$

Hardamard–Rybczynski For a more accurate model the bubble is seen as a fluid sphere moving through a liquid medium. The Hardamard–Rybczynski equation (see [Ryb11]) describes the drag force of a fluid sphere with viscosity η' as

$$F_w = 6\pi\eta RU\frac{2\eta + e\eta'}{3\eta + 3\eta'}. \tag{A.2}$$

Due to the difference in density, a gas bubble in a liquid medium is under the influence of an uplifting force directly proportional to the ratio of the densities of the bubble and the medium. The uplifting force and the drag force pull in oppositional directions. If the uplifting force is larger, the bubble is accelerating its ascend. If the drag force is larger, the bubbles rising motion is decelerated.

In case of an equilibrium the uplifting speed of the bubble is (see [Czi91, p.E136])

$$w = \frac{1}{3}\frac{g}{\nu}R^2 \tag{A.3}$$

where g is gravitational force and v is the kinematic viscosity, assuming that the density ρ' and the viscosity η' of the bubble are significantly lower than the density ρ and the viscosity η of the medium.

Bibliography

[AF95] William E. Asher and Paul J. Farley. Phase-doppler anemometer measurement of bubble concentrations in laboratory-simulated breaking waves. *Journal of Geophysical Research*, 100(C4):7045, 1995.

[AH12] Anne Auger and Nikolaus Hansen. Tutorial CMA-ES: evolution strategies and covariance matrix adaptation. page 827. ACM Press, 2012.

[BB87] S. Baldy and M. Bourguel. Bubbles between the wave trough and wave crest levels. *Journal of Geophysical Research*, 92(C3):2919, 1987.

[BDZ+13] Yuchen Bian, Feng Dong, Weida Zhang, Hongyi Wang, Chao Tan, and Zhiqiang Zhang. 3D reconstruction of single rising bubble in water using digital image processing and characteristic matrix. *Particuology*, 11(2):170–183, April 2013.

[Bel80] Edward O. Belcher. Quantification of bubbles formed in animals and man during decompression. *IEEE Transactions on Biomedical Engineering*, BME-27(6):330–338, June 1980.

[BMCM97] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *, 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings*, pages 495–501, 1997.

[BMKD12] Kausik Basak, Manjunatha M, and Pranab Kumar Dutta. Pyramidal refinement of lucas kanade optical flow based tracking of peripheral air embolism in OCT contrast imaging. *International Journal of Computer Applications*, 52(12):7–12, August 2012.

[Bry33] T. Bryn. Steiggeschwindigkeit von luftblasen in flüssigkeiten. *Forschung auf dem Gebiete des Ingenieurwesens*, 4(1):27–30, February 1933.

[BS84] I. N. Bronstein and K. A. Semendjajew. *Taschenbuch der Mathematik*. Harri Deutsch, Thun, Frankfurt(Main), 21 edition, 1984.

[Buc94] T.D. Bucknell. A novel method for optical bubble-sizing. volume 1994, pages 141–147. IEE, 1994.

[Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[CB01] Da-chuan Cheng and Hans Burkhardt. Bubble recognition from image sequences. *PROCEEDINGS OF EUROTHERM SEMINAR 68, INVERSE PROBLEMS AND EXPERIMENTAL DESIGN IN THERMAL AND MECHANICAL ENGINEERING*, 2001.

[CB03] Da-Chuan Cheng and Hans Burkhardt. Bubble tracking in image sequences. *International Journal of Thermal Sciences*, 42(7):647–655, July 2003.

[CB06] Da-chuan Cheng and Hans Burkhardt. Template-based bubble identification and tracking in image sequences. *International Journal of Thermal Sciences*, 45(3):321–330, March 2006.

[cor09] *Introduction to algorithms*. MIT Press, Cambridge, Mass, 3rd ed edition, 2009.

[CTMM02] H M Choi, T Terauchi, H Monji, and G Matsui. Visualization of bubble-fluid interaction by a moving object flow image analyzer system. *Annals of the New York Academy of Sciences*, 972:235–241, October 2002. PMID: 12496023.

[Czi91] Horst Czichos. *Hütte: die Grundlagen der Ingenieurwissenschaften*. Springer, Berlin, 29 edition, 1991.

[DH92] P.R. Dando and M. Hovland. Environmental effects of submarine seeping natural gas. *Continental Shelf Research*, 12(10):1197–1207, October 1992.

[FF95] Andrew Fitzgibbon and Robert B. Fisher. A buyer's guide to conic fitting. In *In British Machine Vision Conference*, page 513–522, 1995.

[FW07] Susanne Fangohr and David K. Woolf. Application of new parameterizations of gas transfer velocity and their impact on regional and global marine CO2 budgets. *Journal of Marine Systems*, 66(1-4):195–203, June 2007.

[GAE+06] J Greinert, Y Artemov, V Egorov, M Debatist, and D Mcginnis. 1300-m-high rising bubbles from mud volcanoes at 2080m in the black sea: Hydroacoustic characteristics and temporal variability. *Earth and Planetary Science Letters*, 244(1-2):1–15, April 2006.

[GL97] John Y. Goulermas and Panos Liatsis. Novel combinatorial probabilistic hough transform technique for detection of underwater bubbles. volume 3029, pages 147–156, 1997.

[Gre11] Point Grey. Grasshopper2 GS2-FW technical reference manual, August 2011.

[Has01] Yassin Hassan. Full-volume, three-dimensional, transient measurements of bubbly flows using particle tracking velocimetry and shadow image velocimetry coupled with pattern recognition techniques. Technical report, Department of Nuclear Engineering, Teax A&M University, , Teax A&M University, 2001.

[HHV04] N.J. Hepworth, J.R.M. Hammond, and J. Varley. Novel application of computer vision to determine bubble size distributions in beer. *Journal of Food Engineering*, 61(1):119–124, January 2004.

[HK04] Nikolaus Hansen and Stefan Kern. Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242, pages 282–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[IKH+04] Atsushi Ishimatsu, Takashi Kikkawa, Masahiro Hayashi, Kyoung-Seon Lee, and Jun Kita. Effects of CO2 on marine fish: Larvae and adults. *Journal of Oceanography*, 60(4):731–741, 2004.

[JC79] B. D. Johnson and R. C. Cooke. Bubble populations and spectra in coastal waters: A photographic approach. *Journal of Geophysical Research*, 84(C7):3761, 1979.

[JMS92] M. Jamialahmadi and H. Müller-Steinhagen. Effect of alcohol, organic acid and potassium chloride concentration on bubble size, bubble rise velocity and gas hold-up in bubble columns. *The Chemical Engineering Journal*, 50(1):47–56, October 1992.

[JSK12] Anne Jordt-Sedlazeck and Reinhard Koch. Refractive calibration of underwater cameras. In *Computer Vision – ECCV 2012*, volume 7576, pages 846–859. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Kal60] RE Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.

[Kuh55] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, March 1955.

[KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[KYT+97] R. Kikuchi, T. Yano, A. Tsutsumi, K. Yoshida, M. Punchochar, and J. Drahos. Diagnosis of chaotic dynamics of bubble motion in a bubble column. *Chemical Engineering Science*, 52(21-22):3741–3745, November 1997.

[LB05] Ira Leifer and Jim Boles. Measurement of marine hydrocarbon seep flow through fractured rock and unconsolidated sediment. *Marine and Petroleum Geology*, 22(4):551–568, April 2005.

[LCD98] Jos Lelieveld, Paul J. Crutzen, and Frank J. Dentener. Changing concentration, lifetime and climate forcing of atmospheric methane. *Tellus B*, 50(2):128–150, April 1998.

[LDLC03] Ira Leifer, Gerrit De Leeuw, and Leo H. Cohen. Optical measurement of bubbles: System design and application. *Journal of atmospheric and oceanic technology*, 20(9):1317–1332, 2003.

[LdLKHC03] Ira Leifer, Gerrit de Leeuw, Gerard Kunz, and Leo H. Cohen. Calibrating optical bubble size by the displaced-mass method. *Chemical Engineering Science*, 58(23-24):5211–5216, December 2003.

[LHQ+13] Du Libin, Zhang Hao, Yang Qian, Wu Chengxuan, and Wang Xiufen. Extraction of bubble contour in single width. pages 559–561. IEEE, January 2013.

[LK05] Sang Joon Lee and Seok Kim. Simultaneous measurement of size and velocity of microbubbles moving in an opaque tube using an x-ray particle tracking velocimetry technique. *Experiments in Fluids*, 39(3):492–497, July 2005.

[LLZR11] Yan-Peng Li, Yan-Yan Liu, Ting-Ting Zhu, and Xiao-Dong Ruan. Imaging study on bubble size distribution in presence of frother in froth flotation. pages 1191–1194. IEEE, May 2011.

[LYX+06] Wei Li, Kecheng Yang, Min Xia, Dan Tan, Xiaohui Zhang, Qizhong Liu, and Xuanhua Lei. Computation of the scattering intensity distribution for natural light scattered by an air bubble in water. *Journal of Optics A: Pure and Applied Optics*, 8(1):93–99, January 2006.

[Mar63] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963.

[MBB+11] Daniel McGinnis, S. E. Beaubien, Nikolaus Bigalke, Lee D. Bryant, M. Celussi, C. Comici, C. De Vittor, Peter Feldens, M. Giani, and A. Karuza. The panarea natural CO2 seeps: fate and impact of the leaking gas (PaCO2); R/V URANIA, cruise no. u10/2011, 27 july–01 august 2011, naples (Italy)–Naples (italy). 2011.

[MBD88] Philip L. Marston, Stuart C. Billette, and Cleon E. Dean. Scattering of light by a coated bubble in water near the critical and brewster scattering angles. pages 308–316, August 1988.

[MGA+06] D. F. McGinnis, J. Greinert, Y. Artemov, S. E. Beaubien, and A. Wüest. Fate of rising methane bubbles in stratified waters: How much methane reaches the atmosphere? *Journal of Geophysical Research*, 111(C9), 2006.

[MM77] C. Masreliez and R. Martin. Robust bayesian estimation for the linear model and robustifying the kalman filter. *IEEE Transactions on Automatic Control*, 22(3):361–371, 1977.

[MP03] Markus Honkanen and Pentti Saarenrinne. Multiphase PIV method with digital object separation methods. In *Proceedings of the Fifth Symposium on Particle Image Velocimetry*, Busan, Korea, 2003.

[MPG90] Menet, Sylvie, Philippe Saint-Marc, and Gérard Medioni. B-snakes: Implementation and application to stereo. *proceedings DARPA*, 720:726, 1990.

[MSMM90] S. Menet, P. Saint-Marc, and G. Medioni. Active contour models: overview, implementation and applications. In , *IEEE International Conference on Systems, Man and Cybernetics, 1990. Conference Proceedings*, pages 194–199, 1990.

[MWZC11] Zhi-guo Ma, Jiang-an Wang, Si-guang Zong, and Dong Chen. Remote optical sensing of bubble films in water by image processing. pages 1–3. IEEE, May 2011.

[Noc06] Jorge Nocedal. *Numerical optimization*. Springer series in operations research. Springer, New York, 2nd ed edition, 2006.

[OAYT06] K. Ohba, Y. Abe, K. Yamamoto, and T. Tajikawa. New method for discriminating emboli from micro-bubbles in blood flow using ultrasound doppler velocimetry. *Journal of Biomechanics*, 39:S302, January 2006.

[OF03] Stanley Osher and Ronald P Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer, New York, 2003.

[PD00] Nikos Paragios and Rachid Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(3):266–280, 2000.

[PJK04] Jack A. Puleo, Rex V. Johnson, and Tim N. Kooney. Laboratory air bubble generation of various size distributions. *Review of Scientific Instruments*, 75(11):4558, 2004.

[Ryb11] Rybzynski, W. Über die fortschreitende bewegung einer flüssigen kugel in einem zähen medium. *Bull. Acad. Sci. Cracovie*, 1911.

[SvDP12] J. Schneider von Deimling and C. Papenberg. Technical note: Detection of gas bubble leakage via correlation of water column multibeam images. *Ocean Science*, 8(2):175–181, March 2012.

[SvDRG+11] J. Schneider von Deimling, G. Rehder, J. Greinert, D.F. McGinnnis, A. Boetius, and P. Linke. Quantification of seep-related methane gas emissions at tommeliten, north sea. *Continental Shelf Research*, 31(7-8):867–878, May 2011.

[Sze11] Richard Szeliski. *Computer vision algorithms and applications*. Springer, London; New York, 2011.

[TZSB10] K. Thomanek, O. Zielinski, H. Sahling, and G. Bohrmann. Automated gas bubble imaging at sea floor; a new method of in situ gas flux quantification. *Ocean Science*, 6(2):549–562, June 2010.

[VF98] S. Vagle and D.M. Farmer. A comparison of four methods for bubble size and void fraction measurements. *IEEE Journal of Oceanic Engineering*, 23(3):211–222, July 1998.

[WB95] Greg Welch and Gary Bishop. *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill, TR 96-041, 1995.

[WD09a] H Y Wang and F Dong. A method for bubble volume calculating in vertical two-phase flow. *Journal of Physics: Conference Series*, 147:012018, February 2009.

[WD09b] Hongyi Wang and Feng Dong. Image features extraction of gas/liquid two-phase flow in horizontal pipeline by GLCM and GLGCM. pages 2–135–2–139. IEEE, August 2009.

[WS92] Donna J. Williams and Mubarak Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image understanding*, 55(1):14–26, 1992.

[YA96] Luren Yang and Fritz Albregtsen. Fast and exact computation of cartesian geometric moments using discrete green's theorem. *Pattern Recognition*, 29(7):1061–1073, July 1996.