JAKOB DE MAEYER

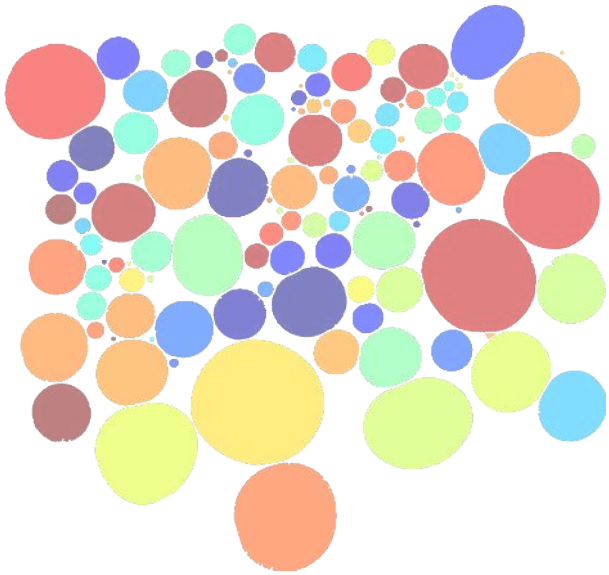
EVOLUTION OF THE SIZE DISTRIBUTION OF OIL DROPLETS ON A WATER SURFACE

SETUP, IMAGE PROCESSING, AND SCALING LAWS

EVOLUTION OF THE SIZE DISTRIBUTION OF OIL DROPLETS ON A WATER SURFACE

SETUP, IMAGE PROCESSING, AND SCALING LAWS

JAKOB DE MAEYER



Master's thesis

Supervised by apl. Prof. Dr. Jürgen Vollmer
and Prof. Dr. Stephan Herminghaus

Principles of Self-Organization group
Department for Dynamics of Complex Fluids
Max Planck Institute for Dynamics and Self-Organization
and
Institute for Nonlinear Dynamics
Faculty of Physics
Georg-August-Universität Göttingen

Submitted 30 September 2015

Jakob de Maeyer:
Evolution of the Size Distribution of Oil Droplets on a Water Surface:
Setup, Image Processing, and Scaling Laws

SUPERVISED BY:
apl. Prof. Dr. Jürgen Vollmer
Prof. Dr. Stephan Herminghaus

AT:
Principles of Self-Organization group
Department for Dynamics of Complex Fluids
Max Planck Institute for Dynamics and Self-Organization
and
Institute for Nonlinear Dynamics
Faculty of Physics
Georg-August-Universität Göttingen

Göttingen, 30 September 2015

ABSTRACT

In this thesis, I describe my work towards a novel experiment to study the dynamics of oil droplets on water. It allows following the evolution of the droplet size distribution during a coalescence process with superb statistics. Every experiment run involves more than 20,000 initial droplets coalescing into a few hundred within four to six hours, with droplet sizes ranging from 0.01 cm^2 to 1,000 cm^2.

Data acquisition is built around an LED light source and a shadow image method that exploits the oil droplets' optical properties. Images of the droplet configuration are taken with a consumer-level DSLR camera and processed with circle detection and morphological methods. I have developed two open-source software frameworks: one facilitates camera remote control at higher frame rates than available in existing software solutions, the other one simplifies the task of batch image processing.

The coalescing oil droplets reveal an uncommon characteristic: As the system evolves, the distribution of droplet sizes becomes bimodal. By dividing the droplets into a group of small and a group of large droplets, we are able to identify three regimes in the coalescence process. We introduce a mathematical model in which the distribution decomposes into a superposition of a steady distribution of small droplets and a scaling distribution of larger droplets.

'Would you tell me, please, which way I ought to go from here?'
'That depends a good deal on where you want to get to,' said the Cat.
'I don't much care where–' said Alice.
'Then it doesn't matter which way you go,' said the Cat.
'–so long as I get somewhere,' Alice added as an explanation.
'Oh, you're sure to do that,' said the Cat, 'if you only walk long enough.'

— Lewis Carroll, *Alice in Wonderland*

ACKNOWLEDGMENTS

I am deeply indebted to Jürgen Vollmer for providing me with his wisdom, equipment, contacts, and, perhaps most importantly, his patience. Everything described in this thesis was formed by the many fruitful discussions we had, and countless times the sometimes unusual, but always brilliant ideas he brought to the table moved it forward when I was stuck. None of this would have been possible without him.

I would further like to thank Stephan Herminghaus for agreeing to the ungrateful task of second reviewer without batting an eye, and more importantly for shaping this wonderful department in which every single person I have met is cordial, bright, and always ready to put his or her own problems aside to help others. Special mentions are due to Kristian Hantke, whom I could always count on to test the usefulness of many wild ideas, and Wolf Keiderling, who never failed to propose experiment designs much more elegant than what I came to him with.

I shall not forget to thank (in no particular order) Martin, Barbara, Johannes, Artur, Bernhard, all of Marco Mazza's group, and everyone roaming in and out of the yellow kitchen, who excelled at creating a welcoming atmosphere and were always there to provide help or distraction, whichever was more necessary. I have very much enjoyed working in this familiar environment.

Finally, I would like to express my gratefulness to the Faculty of Physics Göttingen. Not only have I been provided with an extraordinary quality of education, I have also never felt left alone with both my professional and personal struggles. Without exceptions, I was allowed to learn under great scientists that are interested in my growth as a physicist, and as a person, rather than in my expulsion, something that sadly has become more and more rare in many academic fields.

CONTENTS

1 MOTIVATION

Coalescene, the merging of two or more droplets or particles, is everywhere. It happens on the scale of molecules[1,2] and on the scale of galaxies[3–5]. It plays a crucial role in natural processes, such as rain formation[6–8] or volcanic magmas[9–11], as well as in industrial processes, such as polymer blending[12,13], oil recovery[14,15], or spray combustion[16,17].

Classic coalescence processes are driven by a decrease in surface or line energy.[18] Simply speaking, a molecule deep inside a liquid droplet finds itself in a favourable energy state, as it benefits from the cohesion with its neighbours. In contrast, a molecule segregated to the surface loses roughly half of these cohesive interactions, and hence lives in an unfavourable energy state. This is the fundamental reason that liquids adjust their shape to expose the smallest possible surface area. The surface tension (or line tension in the case of two-dimensional droplets) γ is a measure for energy necessary to uphold one unit of surface area. Minimising the surface or line energy, i. e. the total energy necessary to maintain the liquids shape, is the main motor behind coalescence: A sphere with volume V has an area about one third smaller than two spheres with Volume $V/2$.

There are many phenomena associated with coalescence that are not yet fully understood. Thoroddsen and Takehara[19] and Blanchette and Bigioni[20] used high-speed video recording to show that the coalescence of a droplet placed onto a layer of the same fluid is not instantaneous, but takes place in a self-similar *cascade*. Gau and Herminghaus[21] and Lapp[22] demonstrated that the dynamics in breath figures, i. e. droplets condensing on a substrate, are dominated by cascades of droplet coalescences. Weon and Je[23] and Kim et al.[24] show that droplet and bubble coalescence is subject to *coalescence preference*, where a bubble merged from two differently sized parent bubbles tends to be placed closer to the larger parent bubble (in agreement with the surface energy release theory), and that the coalescence preference is inhibited in densely packed clusters of microbubbles.

Moreover, coalescence has fields of application that are not obvious at first sight. The very first ideas on the way to this thesis stemmed from the study of *explosive percolation*. Percolation theory is the study of the number, properties, and behaviour of connected clusters in random networks.[25] While a long-standing topic in the field of statistical physics, going back to the late 50s[26], recent advances by Achlioptas et al.[27], da Costa et al.[28,29], Nagler et al.[30], and Riordan and Warnke[31,32] (among many others) have inflamed a lively discussion

on the existence and the prerequisites necessary to observe a macroscopic gap in the percolation function, i. e. a jump in the fraction of network nodes that are part of the largest existing cluster under the addition of links, even in the limit of infinitely many network nodes. In a coalescence experiment, the individual droplets or bubbles can be thought of as clusters, and a merging event corresponds to adding a link to the network that connects two clusters.

Single coalescence events are often studied under microscopes[33–36] or, more recently, via x-ray imaging[23,24]. Owing to its many applications in industrial fields, the distribution of droplet sizes is typically investigated in fragmentation rather than coalescence processes[37,38], with the help of atomisers,[39,40] such as fuel injection systems for combustion engines, or strong ultrasound waves[41]. The droplets and bubbles available through these methods have diameters in the range of tens or hundreds of micrometers.

However, results from experiments with larger droplets, as well as studies on the *evolution* of the droplet size distribution under the influence of coalescence, are scarce. This hinders experimental insight into phenomena that happen in the *tails* of distribution functions, such as the evolution of the largest cluster in a percolating system, or the initiation of rain.[42]

1.1 OUTLINE

In this thesis, I introduce a novel experiment that uses oil droplets on a water surface to extract high-quality data on the time evolution of droplet size distributions in a coalescence process. I follow a bottom-up approach: First, chapter 2 describe the thoughts that went into the design of the experimental setup, and the advances and setbacks that lay along the way to build it. Next, chapter 3 addresses the task of acquiring valuable data from the experiment; namely, how do we capture images from the experiment, and how can we automatically process them. Last, chapter 4 uses the obtained data to learn about specific traits of our experiment, and about features of its coalescence behaviour.

The main goal of this thesis is to completely describe, characterise, and generically evaluate a new physical experiment. Hence, it aims at providing the reader with the information necessary to understand the experimental setup as well as the software used for image acquisition and analysis, including the image processing algorithms. I do not go into an evaluation aimed at specific topics or applications of coalescence, such as the coalescence preference or explosive percolation. Rather, I provide a general analysis of the experiment in order to gain insight into the statistical physics underlying its coalescence process.

2 EXPERIMENTAL SETUP

Hofstadter's Law: It always takes longer than you expect,
even when you take into account Hofstadter's Law.
— Douglas Hofstadter[43]

Albeit short, this chapter describes the basis of this thesis: The experimental setup built to provide statistics on the coalescence of oil droplets on a water surface. I will start with an anecdote on the inspiration for the experiment before presenting the fundamental requirements that it was built against. With these in mind, section 2.2 retraces the many adventures and misadventures that lay along the way to building the experiment. Finally, the current setup is summarised in section 2.3, and I describe the procedures that were followed for every run of the experiment in section 2.4.

Before we start, it seems in order to clarify on a nomenclature issue: Many authors[44–47] use the term *oil lenses* for small oil droplets on a surface. It is applicable to their cases since their oil droplets, mostly examined in a microscope, are indeed shaped like circular lenses. As we will see later, the larger oil droplets we encounter in the experiment, far larger than what can be viewed through a microscope, are more pancake- than lens-shaped, and I will hence stay with the term *oil droplet* throughout this thesis.

2.1 FROM SALAD DRESSING TO EXPERIMENT

The salad at the *Zentralmensa* in Göttingen often comes with a mediocre, standard dressing, made mostly of water, oil, and vinegar. Probably familiar to the reader from cooking soups, the oil forms coalescing droplets on the water surface. During a long discussion over finished plates, the following can be observed: through evaporation, the water level in the bowl continuously decreases. Due to the shape of the bowl, this reduces the dressing's surface area. The oil droplets are thus forced to share a smaller area, which accelerates the coalescence process. When the water has fully evaporated, we find a single, large oil droplet.

The principal ambition of this thesis was to transfer the phenomenon of the coalescing salad dressing into a controllable, reproducible laboratory experiment. To allow quantitative statements, great effort must be put towards fast and extensive data acquisition. This imposes the following requirements on the experiment:

(i) *Large possible numbers of oil droplets* allow exploring a great range of size distributions, and furthermore play a crucial role in the variation of system size to facilitate finite size scaling arguments.

(ii) *Short experiment times* are a key ingredient in achieving a reliable level of statistics through many experiment repetitions, therefore reducing errors in quantitative measurements.

(iii) *High-resolution imaging, both spatially and temporally*, dictates the range of droplet sizes that can be explored.

(iv) *Reliable droplet detection* sets the standard for both accuracy and precision of the quantitative measurements.

Satisfying the latter two requirements is the heart of chapter 3, while accomplishing the former two is discussed in the remainder of this chapter.

2.2 A LONG WAY...

This section contains what is usually left out in a thesis: All the failures, distractions, and 180 degree turns that precede what is finally achieved and presented as straight-forward. I decided to include it since I spent an unsatisfactorily large chunk of my time on working around the many stumbling blocks. The following paragraphs serve as a documentation of my work, and I hope that they prove useful when discussing further improvements and enhancements of the experiment. They are intended as a collection of hints for a potential student picking up the experiment later. As such, to not unnecessary bloat this thesis, they contain a summary of my experience *rather than an extended analysis of all the small experiments that lay along the way. What follows is not necessary for understanding the final experimental methods and materials, nor the results, and the incurious or impatient reader should feel free to skip to the next section.*

Canonically, the components of an experiment are introduced thematically. However, since this section describes the history *of the experiment, it follows the experiment construction* chronologically.

2.2.1 *Prelude*

A group of students from the University of Göttingen, Happ et al.[48], tried to evaluate the coalescing salad dressing in a short laboratory (*Projektpraktikum*) during the summer semester of 2011. They poured 20 ml of water with one percent soap and 25 droplets of pumpkin seed oil into a watch glass. Images of the merging processes during water evaporation (about 24 hours) were taken with a low-resolution video camera and processed with threshold and edge detection algorithms. Their main hope was to review and provide support for

numerical results on critical exponents in competitive percolation[49] by Nagler et al.[30]. Unfortunately, their primary result was that a much larger system size is needed to make quantitative statements, with their system size insufficient to determine any of the exponents they were interested in to a satisfying degree.

There are two ways to increase the initial number of oil droplets: reducing the (initial) size of the droplets, and increasing the area on which they (initially) live. I opted to do both, and dealt with increasing the geometrical size of the experiment first. The watch glass was replaced with a large funnel of about one meter in diameter and half a meter in height. Naturally, waiting for the corresponding mass of water to evaporate collides with our requirement (iii) (short experiment times). Hence, the funnel was equipped with an outlet at its bottom, allowing the experimenter to set the water level's decrease rate at will.

2.2.2 *Making the droplets visible*

Happ et al.[48] used a piece of white paper underneath the watch glasses to increase the contrast between background and oil droplets. Buying a custom-made glass funnel of this size is, however, unreasonably expensive (if possible at all). The metallic surface of the funnel significantly reduces the droplets-to-background intensity ratio, and its reflectivity messes with the intensity profile of the background illumination. While this can be dealt with by image post-processing and painting the funnel, there is a still bigger problem: As the oil droplets are rather thin layers, their visibility is strongly dependent on their height. Using only their light absorption to differentiate them from a white background unnecessarily increases the lower bound of detectable droplet sizes, and therefore contradict our requirements (i) (large possible numbers of oil droplets) and (iv) (reliable droplet detection).

With this in mind, and in the naïve hope of a near-perfect signal-to-background ratio, I turned to fluorescence. The pumpkin seed oil was replaced with hexadecane and a lipophilic, bright, and photostable fluorescent marker, Nile Red. A spatially (relatively) homogeneous excitation over the complete area of the funnel can be achieved with an array of bright light-emitting diodes. Their dense emission spectrum would leave the emission spectrum nearly unpolluted (cf. fig. 1). The camera must then be equipped with a corresponding low-, or better yet, bandpass filter. Dishearteningly, fluorescence tests in a microscope showed that despite Nile Red's high quantum yield[50], the fluorescence from a thin hexadecane layer is too weak: At an illumination intensity higher than achievable with the bright LED array, and an exposure time of one second (!) (the aperture being predetermined by the microscope), a DSLR camera similar to the one we planned to

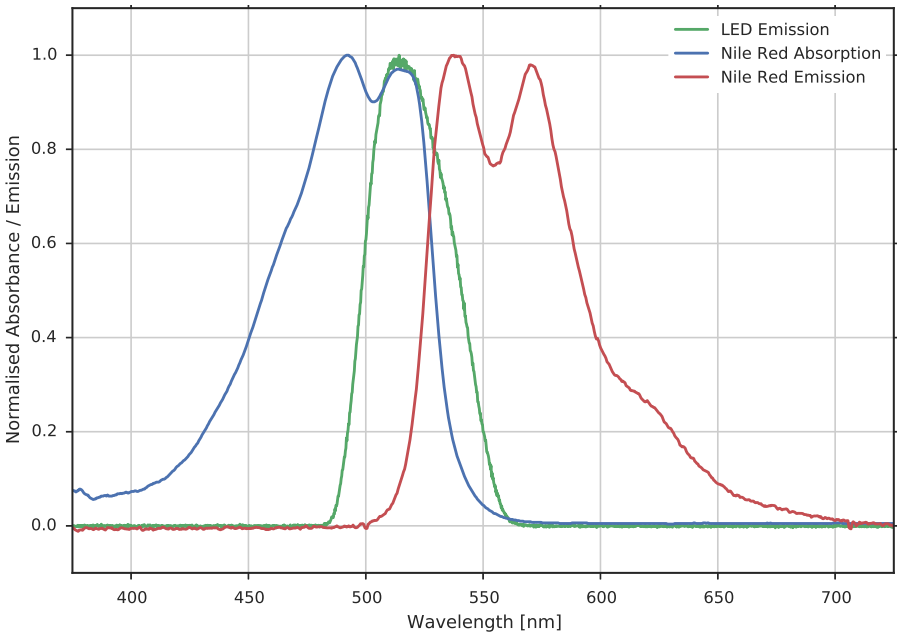


Figure 1: Fluorescence and LED spectra. Nile Red's absorption spectrum in hexadecane (measured at Kris Hantke's lab at the MPIDS) shows two peaks. The second of these peaks has a near-perfect match with the typical emission spectrum of a green LED (data shown is for a Thorlabs M530L3), while the corresponding Nile Red emission peak has little to no overlap with the LED.

use in our experiment picked up the fluorescence peak at 18 of the 255 available arbitrary intensity units, above a base noise level of 3.

By now, oil absorption and fluorescence had been ruled out as methods to visualise the oil droplets for image processing. In search of a new method (and a hint of desperation), during a discussion in the institute's infamous *yellow kitchen*, we filled water and oil into a casserole and played around with an LED flash light, trying to figure out how our own eyes distinguish the oil droplets from the water surrounding them. After establishing that our eyes are not too good at seeing the droplets after all, we noticed that with light coming in perpendicular (i. e. from the top), each droplet produces a shadow image of its shape (cf. fig. 2), with a very stark contrast between the bright center of the droplet and the almost completely dark edge, where the droplet's curvature bends the light away.

There are two major obstacles on the way to implementing the shadow-image method in our funnel setup. First, as the path of the light towards the bottom is obscured by the non-transparent funnel, there is no possibility to visualise the shadows with a light source set up above. This can be resolved by placing a watertight light source with wide opening angle at the bottom of the funnel, and by placing a screen at its top. Second, and more grave, the shadow image of the oil droplets is strongly dependent on their distance towards

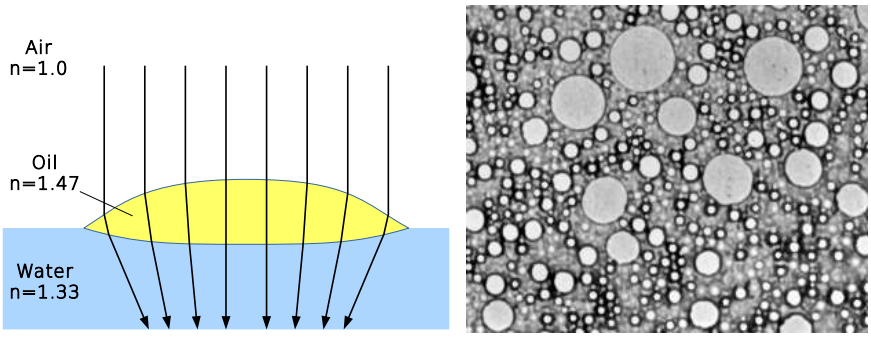


Figure 2: Droplet shadows. *Left*: Due to the curvature of its edge[46], and the different refractive indices of the three materials involved[51,52], parallel light falling onto an oil droplet on a water surface creates a shadow image with a dark outline of the droplet's edge and a bright center. *Right*: Droplet shadows in an image from the final experiment.

the screen. With growing distance, the shadows become increasingly blurred. While this can be computed back in post-processing (it is linear optics, after all), the effort would be huge, and would have exceeded the scope of a master's thesis.

Mastering the latter obstacle without excessive post-processing calls for a constant distance between water level and screen. Given that there is no obvious way to (i) keep the screen as close to the water level as possible, (ii) make it follow the water level during an experiment and at the same (iii) constantly change its size so that it fits into the funnel at any given height, there were two possible choices at this point: abandoning the shadow-image method, or committing to a constant water level, essentially rendering the shape of the funnel, and its outlet at the bottom, useless.

The shadow images were so temptingly beautiful, with much higher contrast and much higher detail than anything we had seen before, that I opted for the latter option and moved to different forms of water basins. A small aquarium served as playground for preliminary tests before upscaling the shadow-image method. The aquarium was filled with water almost to the top, where a white sheet of paper served as screen. A number of light sources were then placed underneath the aquarium.

2.2.3 *Illumination*

As Lucas Goehring once put it, "image processing on inhomogeneous backgrounds is a bitch." For ease of post-processing, the light source for our experiment should be homogeneous on a large area. At the same time, to use the lens properties of the oil and end up with a sharp shadow image, it needs to produce collimated light. Several

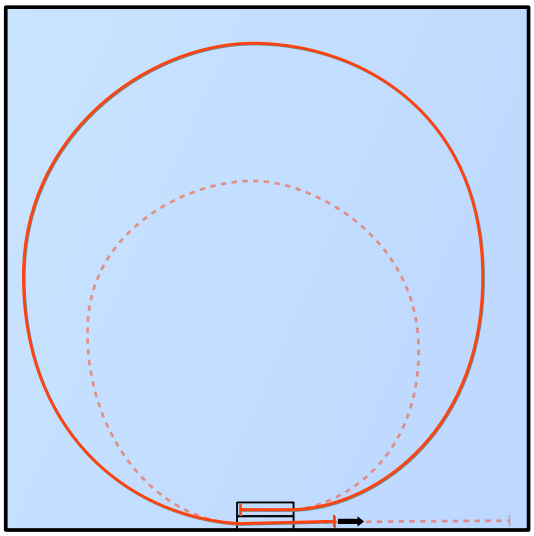


Figure 3: Controlling the effective surface area with a floating tube. The oil droplets, trapped within the boundaries set by the tube (red, solid), can be pushed together by pulling on one end of the tube (dashed). The process can be automated and smoothed by using a small electric gear motor.

light sources that are typically used for large-scale illumination, such as halogen floodlights with a parabolic mirror or LED illumination screens, were quickly ruled out. The missing collimation makes the shadow images they produce too blurry. On the other end of the spectrum, dedicated collimated light sources are only available with small beam diameters of up to 50 mm. Finally, we ended where we started: Flashlights that have an LED chip and a built-in lens behave like a point source: At larger distances, the light rays can be considered *locally collimated* (for the typical size of an oil droplet); and while the background illumination is by no means homogeneous, its heterogeneity is at least somewhat well defined.

2.2.4 *Controlling the surface area*

Having ensured the practicability of the shadow-image method, we now focus on the problem that was deliberately swept under the carpet in section 2.2.3: controlling the surface area on which the oil droplets live. The rectangular shape of a typical aquarium forbids using evaporation or water draining to continuously drive the droplets closer together. Kristian Hantke, who had also previously helped with (and was rightfully sceptical about) the tests on fluorescence, suggested using a floating tube. The tube is laid out on the water in a loop, such that the radius can be controlled by pulling (or pushing) on one end (cf. fig. 3). While the oil, bound to the surface, is trapped inside the shrinking loop, the water is free to move underneath. A

small electric gear motor, as commonly used in model making, can be used to control and stabilise the rate of surface area reduction. While the implementation involved a few trips to aquarium and hobby centers, as well as learning some new skills in the mechanics of holding tubes and in electronics, first working versions of this method turned out to be promising.

2.2.5 *Restoring the funnel's large size*

While a big step up from the watch glass, the test aquarium with its size of about 20 cm $\times$ 40 cm still had a lot of room for improvement towards our requirement (i) (large possible numbers of oil droplets). A custom-made aquarium with a base area of 1 m $\times$ 1 m, and a height of 10 cm, was bought. Upscaling most other parts of the setup was unproblematic: the LED flashlight was still sufficiently bright even from an increased distance (for a larger area of illumination), and the motor had to be replaced with a stronger version. The screen was placed underneath the aquarium, squeezed between its bottom and the top of an acrylic glass, as it would otherwise bend in too much at the center of the aquarium.

Unexpectedly, however, there were major complications with the tube that was supposed to push the droplets closer together. Even though it is made of plastic and filled with air, short segments of it sank below the water surface, thereby producing a leak for the oil droplets and effectively cancelling the reduction in surface area. Attaching small water wings made of bubble wrap to parts of the tube only moved the leaks around, but never closed all of them. At the same time, attaching bubble wrap to the complete tube was not possible, as the modified parts of the tube were too big for the tube mounts. At this point, to move on with the thesis, I abandoned the tube method and went on to an approach described in the next paragraph. However, I believe with more time to explore different tube materials, helpers, and mounts, the tube method can be made to work, and possibly facilitate better control over the coalescence behaviour than available with the current setup.

2.2.6 *Defining initial conditions*

In parallel to all the misses and successes described before, I spent a lot of time in the chemistry lab, searching for ways to influence initial conditions and the speed of the coalescence process. Initially, I had planned to spray oil onto the water surface, as this would produce very small initial oil droplets (working towards requirement (i)). However, with the oil atomisers available on the consumer market, there is very little control over the amount of oil deposited into the experiment. It should also be noted that *inhaled oil will form thin lay-*

ers on the alveoli in our lungs, much like on water, and lead to suffocation despite breathing. Working with an appropriate mask is therefore essential.

To make different experiment runs comparable, I switched to defining the amount of oil in the experiment as a starting point: Instead of spraying oil onto the water, I pour a well-defined, carefully measured amount of oil onto the water. The resulting single large oil droplet can then be split into many small ones by stirring the water-oil mixture with a spoon. The small droplets will relax into a circular shape faster when a surfactant is added to the water, for which sodium dodecyl sulfate (SDS) was used.

After a few tries, it became clear that the speed of the coalescence process is mainly governed by the amount of oil used. While runs with small amounts of oil did show few coalescences even in a time frame of hours, the oil droplets formed with large amounts were hard to even split into small droplets, and coalesced into a large one within seconds. The quantitative values of *small* and *large* naturally vary with the size of the setup. In the petri dish used in the chemistry lab, 20 cm in diameter, they would correspond to 10 ml and 30 ml. In between these two extremes is an amount of oil that seems just right for our purposes: At around 22 ml, the large oil droplet poured in at the beginning can easily be split into many small ones by stirring for about one minute. Afterwards, the (left alone) droplets start coalescing until they form one big and few small droplets after 45 to 60 minutes.

This procedure can be transferred to the big aquarium with almost no modifications. Despite the roughly 30 times larger area, the optimum amount of oil increases only by a factor of nine. After the initial stirring, still about one minute, the droplets start a coalescence process that ends in one large and several smaller droplets after about four hours, without any external influences on the droplets (i. e. without the plastic tube). In the first seconds and minutes, the coalescences happen both fast and on a small (millimeter) scale, further manifesting requirement (iii) (high-resolution imaging).

2.3 MATERIALS

Figure 4 shows the experiment in its current incarnation. It consists of:

- Aluminum frame made of 40 mm × 40 mm BLOCAN structural profiles by Rose+Krieger GmbH (Minden, Germany), total height 220 cm
- Custom-made glass aquarium, 1 m × 1 m × 10 cm, by AQUA SCHWARZ GmbH (Göttingen, Germany)

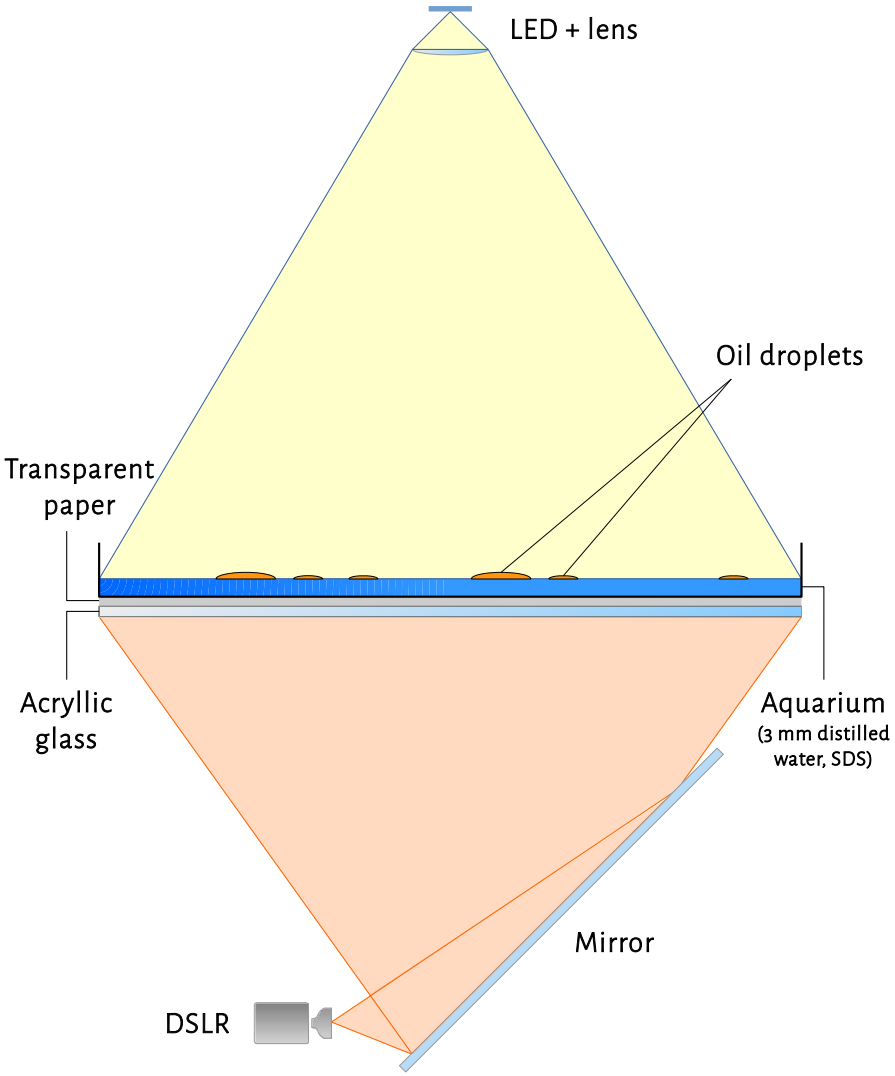


Figure 4: Experimental setup. Oil droplets on a water surface are illuminated with locally collimated light from an LED source above. Their shadows, high-contrast outlines of the droplets due to how they bend light, are cast onto a transparent paper just beneath the aquarium which holds the water and oil droplets. A digital single-lens reflex camera (DSLR) takes high-resolution images of the shadow figures.

- Transparent paper, as used for engineering drawings before the days of CAD (computer-aided design)
- Acrylic glass plate, 1 m × 1 m × 5 mm
- Flashlight with a white Cree (Durham, North Carolina, USA) Q3 LED and lens, by Fox Outdoor (Melrose Park, Illinois, USA), modified by the MPIDS electronic workshop such that it runs from a power supply instead of a battery
- Mirror (customary), 140 cm × 60 cm
- D90 digital single-lens reflex camera, 4,288 × 2,848 pixels, with an 18 mm wide-angle lens, by Nikon (Chiyoda, Japan)

Near-parallel light emerges from an LED at the top of the setup and falls onto the oil droplets, which sit on a water surface in a 1 m² aquarium. The curvature at the droplets' edges bends the light inwards, such that each droplet throws a shadow in the form of its outline. These shadows become visible on a transparent paper just below the aquarium, and are picked up by a camera equipped with a wide-angle lens. Due to the restricted space, the optical path length is increased with a mirror angled at 45°.

2.4 EXPERIMENTAL PROCEDURE

All the experiment runs that are evaluated in chapter 4 were prepared in exactly the same fashion. First, the aquarium is cleaned from the previous run (or from dust gathering in it over weekends, etc.):

(i) Remaining water is drained off through siphoning

(ii) Remaining oil is removed as good as possible with paper towels

(iii) The aquarium is cleaned with ethanol and paper towels, then rinsed with distilled water

(iv) The aquarium is cleaned with acetone and paper towels, then rinsed with distilled water

(v) The aquarium is rinsed with distilled water

(vi) When necessary after visual inspection, the last three steps are repeated

With a clean aquarium, an experiment run is prepared as follows:

(i) Three liters of distilled water are filled into the aquarium (resulting in a water level of 3 mm)

(ii) As surfactant, 2 milliliters of a water-SDS (sodium dodecyl sulfate) solution, with 10 % SDS by weight, are added to the water in the aquarium, which is subsequently stirred

(iii) When the water has settled, 200 milliliters of olive oil (Frantò Natives Olivenöl Extra, purchasable at Kaufland) are carefully poured onto the water surface, such that it forms a single large oil droplet

Next, the experiment run is initiated:

(i) The image capturing software (see chapter 3) is set to a countdown of one minute

(ii) As soon as the countdown is started, the setup is stirred with a plastic shovel

(iii) When the first picture is taken (i. e. when the countdown is over), we stop stirring the setup

At this point, the experiment is left untouched for at least four hours, with pictures taken at subsecond intervals (starting at 4 fps) for the first few minutes, and then gradually fading into larger intervals (up to one minute) as the coalescence speed decays. Images are taken with a 1/15 second exposure time at an f/4.5 aperture and ISO 400, providing a reasonable trade-off between depth of field (to compensate for a slightly off focus plane if necessary), short exposure, and noise level.

3

IMAGE CAPTURING AND PROCESSING

A large portion of the work leading to this thesis comprised the development of two software projects, one for image acquisition and one for image processing. While I always had the experiment in the back of my head, I hoped that they may be useful for other experimentalists at the institute. Hence, they were designed as frameworks, with as much flexibility as possible. Both software projects are distributed as packages for the Python programming language, such that they can be easily configured and used from a user-friendly scripting language, or integrated into other Python projects. To further ease usage, they come with a command line interface.

The first software project, *ICE* (*Image Capturing for Experimentalists*), facilitates remote-controlling digital cameras. It has two major advantages over other available software solutions: First, it is able to use the camera's *burst mode* during remote control, allowing full-resolution frame rates about ten times faster than other software solutions. Second, it gives users full flexibility in choosing the image capturing intervals during time lapses, instead of pinning them down to a simple "*take X pictures, one every Y seconds*" scheme.

The second software project, *FIRE* (*Framed Image Recognition for Experimentalists*), is a small framework that provides boilerplate† for digital image processing, releasing users from having to come up with their own workflow (and, too often, immortalising it in inflexible and hard-to-maintain code). Instead, image processing algorithms are wrapped in modular units which can be selected, ordered, and configured from a human-readable, easily maintainable configuration file.

3.1 EXPERIMENTAL IMAGE CAPTURING

When choosing an image capturing solution for an experiment, you get to pick two out of three attributes: high-resolution, fast, affordable. Simultaneous imaging of tens of thousands of oil droplets requires a high resolution. However, as we will later see, the experimental system generates many merging events within the first few

† In programming, *boilerplate* is code that can be used in many applications and contexts with little to no alteration. The term originates from the newspaper industry in the early 1900s, where advertisements or syndicated columns were sent to subscribing newspapers as a mat, and boiling lead needed to be poured into the mat to create the plate used to print the piece.[53] As these articles printed on a boilerplate could not be altered, the term became used more widely for portions of language that did not change while used for different occasions.

seconds, making a high *time* resolution crucial as well. Here, we present a solution to this problem that avoids buying a high-speed, high-resolution camera.

The Department for Dynamics of Complex Fluids (*DCF*) at the MPI-DS made good experiences with entry-level digital single-lens reflex cameras (*DSLRs*) in the past. These offer high resolutions, typically between 10 and 25 megapixels, and high flexibility through their interchangeable lenses, while being both cheap and readily available due to their wide distribution on the consumer markets. Their drawback, typically, is that they are slow. The data for this thesis was taken with a Nikon D90, a DSLR featuring a 12.3 megapixel CMOS sensor that sells between 500 and 600 € (excluding VAT) in a kit with a wide angle zoom lens.

3.1.1 *The problem with available remote control software*

There are several software solutions available for remote-controlling (*tethering*) Nikon DSLRs, such as *DigiCamControl* (open source), *NKRemote* (commercial), or Nikon's own *Camera Control Pro* (commercial). All of these solutions gather pictures from the camera in the same fashion: First, they trigger the camera. The camera then goes through its internal picture-taking process (i. e. close down the aperture, flap up the mirror, etc., until the sensor has been read out) without any remote interaction. The picture is saved to the camera's internal SDRAM buffer, from which the software transfers it onto the remote-controlling computer via USB. This last part is the major bottleneck for high-speed image acquisition: downloading a full-resolution raw image (roughly 10 MB) takes about three seconds due to the cheap USB chips used in entry-level DSLRs. This limits the effective frame rate for taking image series with the available software solutions to 1/3 frames per second (fps).

However, the camera is technically able to achieve much higher frame rates. When hand-controlled, setting it to *burst mode* allows (continuous) frame rates of up to 4 fps, i. e. a factor ten faster than available during remote control. The difference stems from avoiding the USB transfer: instead of transferred to the remote computer, the pictures from the internal buffer are saved on an SD card plugged into the camera. Since the frame rate in hand-held operation is often a major selling point, camera manufacturers tend to spend money on a fast SD card bus, rather than on the USB chip.

While four frames per second is not incredibly fast, it is impressive when considering the high resolution and the low price, and a vast improvement over the frame rates available through tethering. Still, hand-controlling a camera for the course of a multi-hour experiment is impractical. A part of this thesis was therefore the development

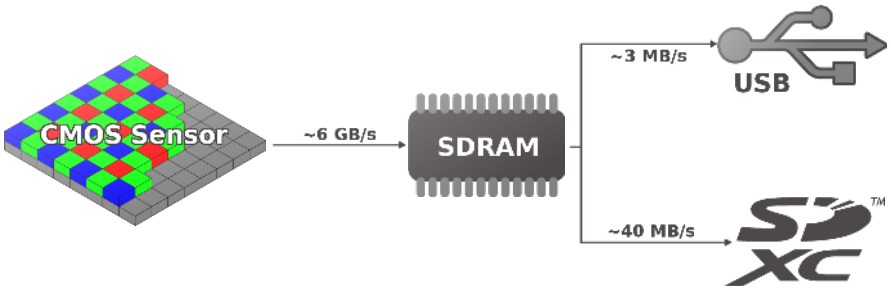


Figure 5: Data flow in a digital camera. Image data is read from the APC CMOS sensor into the camera's internal SDRAM buffer. Unlike other remote control software ICE retains the possibility of using the camera's highest available frame rate at full resolution by saving pictures from the buffer to an SD card instead of transferring them via USB. Bus transfer rates are rough estimates from experience with a Nikon D90 (SDRAM rate from data sheet).

of a software solution that uses the camera's full potential without requiring any human interaction during the experiment.

3.1.2 *Introducing fast and flexible camera tethering with ICE*

The basic idea of the first software solution, *ICE* (*Image Capturing for Experimentalists*), aimed at achieving high-resolution, fast, remote-controlled image acquisition, is very simple: While remote-controlling the camera, mimic its behaviour from hand-controlled mode. That is, make the camera save pictures to an SD card instead of transferring them to the computer. ICE also provides another simple feature not available in other solutions: variable frame rates. While the coalescence experiment changes fast at the beginning, it becomes increasingly slower with time. Hence, images should be taken at small intervals early on and larger intervals during later stages to avoid camera wear and huge data sets.

Almost all modern cameras that have a USB interface implement the Picture Transfer Protocol (*PTP*), a standard that facilitates remote-controlling as well as tranferring data to and from *still image capture devices* (i. e. cameras). For the Linux operating system, there is a stable and well-maintained library, *libgphoto2*[54], that implements PTP and includes many small improvements and hacks to unify the small differences in the remote control interfaces of different manufacturers (for DSLRs, these are mainly Nikon and Canon). ICE is a Python package based on bindings for this library.

3.1.2.1 *Fast frame rates via remote-controlled burst mode*

So what is the black magic in ICE that allows much higher frame rates than other tethering solutions? Libgphoto2 provides two main calls for taking still images: `capture-image` and `trigger-capture`. The

former results in the behaviour found in available software solutions: One or (depending on the camera's settings) multiple pictures are taken, then transferred from the internal buffer onto the tethering computer via USB. During this transfer, the camera is unresponsive to further library calls. The second call, `trigger-capture`, works on a much lower level. It does roughly the equivalent of pushing the shutter release button: trigger the camera's internal picture-taking process, then hand back control immediately. In particular, there is no automatic USB transfer. Instead, the image either remains on the camera's internal buffer or is saved to the SD card, depending on the camera configuration. While the library can reconfigure the camera or send new commands with minimal delay, it is crucial to wait for the camera to signal that the picture has been saved (to the buffer) before triggering the next capture process.

ICE uses libgphoto2's ability to remotely set a camera's configuration, enabling burst mode and setting the desired frame rate and number of pictures to be taken, then starts image acquisition via `trigger-capture`. Although this approach is very simple, it sets ICE apart from other software solutions, allowing remote control of DSLRs without sacrificing their fast burst mode frame rate. The drawback of this method is that pictures are saved to an SD card instead of a computer hard drive, and therefore

(i) the SD card needs to be large enough to hold *all* pictures of an experiment run,

(ii) the SD card needs to be fast enough for the desired frame rate,

(iii) pictures cannot be evaluated immediately.

The impact of (iii) must be evaluated from use case to use case, and does not pose a problem for the experiment described in this thesis. And while (i) is typically not a problem, with 64 GB and 128 GB SDXC cards readily available these days, requirement (ii) touches the limits of many SD cards. A raw, 12 megapixel camera picture has a size of about 10 MB. Sustaining a frame rate of four frames per seconds consequentially requires a write speed of 40 MB/s. There are many SD cards that support such fast writes, especially these with a UHS-I interface.[55] Many older or cheaper cards have write speeds around 20 MB/s. For these, either resorting to two frames per seconds (reminder: the frame rate available with USB transfer is around one frame every three seconds), or accepting minimal image quality loss by enabling the camera's high-quality JPEG compression, will usually do the trick.

3.1.2.2 *Flexible time lapse configuration*

ICE's second feature is the freedom to set variable frame rates. Available software solutions are not aimed at experimentalists. They allow

studio photographers to put their camera on a tripod, then remotely change its exposure settings and capture images directly to a computer, where a large screen for viewing and post-processing tools are immediately available, and they allow hobby photographers to take picture series of their garden or a sunset. Hence, their time lapse configuration is usually limited to "*Take X pictures, one every Y seconds*", or "*Take a picture every Y seconds until Z*". As the tethering software simply translates these settings into "*trigger image capture, then wait Y seconds (in software); repeat X times*", providing greater flexibility is not hard. ICE provides the user with two ways two configure time lapses:

(i) provide a list of millisecond timestamps when images should be taken, e.g. `[0, 250, 500, 750, 1000, 2500, 5000, 10000]`, or

(ii) provide a string with "number of pictures @ frame rate" instructions, e.g. `"4 @ 4fps, 1 @ 1.5s, 1 @ 2.5s, 2 @ 5s"` for the same picture timestamps as above.

In both cases, ICE will internally determine list of camera configuration and trigger commands, together with a list of timestamps when they should be sent. For frame rates faster than one frame per second, the camera needs to be set into burst mode. For slower frame rates, it is sufficient to set the camera to single shot mode, then trigger the capture and wait (in ICE) until the next trigger command needs to be sent.

3.2 PROCESSING EXPERIMENT IMAGES

Computer vision, i. e. the field of recovering information about a real-world scene from its two-dimensional projections,[56] and *digital image processing*, i. e. the field of transforming images into other images in order to enhance particular information and suppress noise, have been around almost since digital computers became available to the scientific community. In 1966, Marvin Minsky at MIT assigned an undergrad student with a summer project (!) "linking a camera to a computer and getting the computer to describe what it saw".[57] At the time, many pioneers in artificial intelligence believed solving the computer vision problem would be an easy step along the path to solving larger problems such as higher-level reasoning. We know now that the problem is *slightly* more difficult than that. A brief, but well-written and extensively illustrated history of computer vision is provided by Szeliski.[57]

Similar to the last section, we first pinpoint some issues that are common in image processing in experimental physics in section 3.2.1. Subsequently, section 3.2.2 shows how FIRE tries to overcome these issues. Sections 3.2.3 to 3.2.5 then introduce the image processing

algorithms and methods used for evaluating the images we gather from our experiment, with the goal of obtaining a list of all droplets, one list per image, with the droplets' size, shape, and position. We employ different algorithms based on the size of the droplets to be detected: For small droplets, we make use of their almost perfectly circular shape by finding pairs of opposite gradient vectors in the image. For larger droplets, whose shapes tend to be less consistent, we turn towards thresholding, morphological operations, and a variety of filters. Finally, section 3.2.6 describe the parameters used for all the different image processing algorithms, and how they were obtained.

3.2.1 *The problem with image processing in experimental physics*

Often, computer vision solutions for acquiring data from images in experimental physics are extremely inflexible. There is a causal chain behind this:

(i) Physicists are typically neither interested in nor experienced with computer vision. Instead, they treat it as a means to extract data from large sets of raw images from an experiment.

(ii) However, computer vision problems often have very specific solutions. Take the experiment described in this thesis: in front of an inhomogeneous but steady background, the size of vastly different oil droplets need to be detected from a shadow image. As we will see below, this requires applying many different algorithms and parameters for different droplet sizes. This is typical for the evaluation of physical experiments with computer vision methods: While the algorithms are readily available in software libraries*, their usage is drastically different from case to case. This forces experimental physicists to write their own code.

(iii) But: While sometimes brilliant in the design and implementation of algorithms, more often than not, physicists have little expertise in designing program architectures.

Consequentially, image processing in experimental physics is often done in badly designed scripts that are extremely inflexible, with hard-coded algorithm calls and parameter values, sometimes even filenames. These scripts solve exactly one measurement or detection problem, and changes to the image evaluation process, even simple parameter adjustments, become tedious and error-prone.

My work for this thesis included writing software that would extract information about the oil droplets out of images from the experiment. However, in order not to end up with yet another static

* The list of available and free image processing libraries is huge. For Python alone there is OpenCV[58], SimpleCV[59], mahotas[60], scikit-image[61], scikit-learn[62], or ilastik[63], to name just a few.

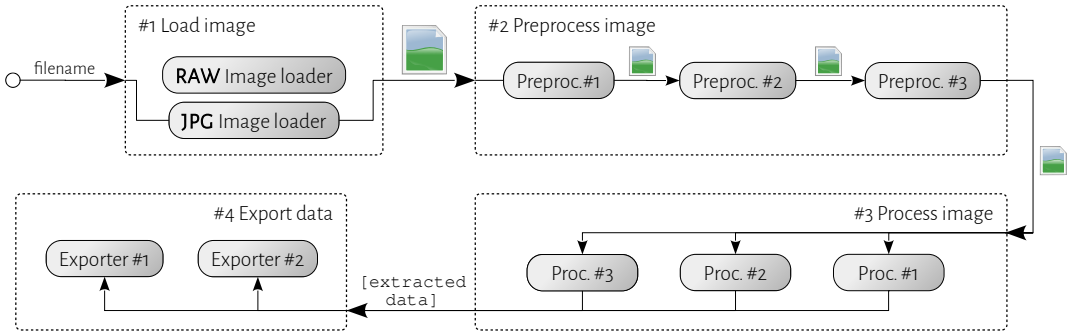


Figure 6: Data flow in FIRE. For each image file to be processed, FIRE selects an appropriate image loader based on file type. The loaded raw image is then given to a chain of preprocessors. Once fully preprocessed, the image is fed to the processors, which return lists with data extracted from the image. Finally, exporters receive this list and save the data to files or a database.

script, I designed *FIRE* (*Framed Image Recognition for Experimentalists*), a framework that facilitates organised image processing. It should be stressed that FIRE is not an image processing library – it is a *framework* that provides the boilerplate for calling algorithms from *other* libraries. Its main feature is a flexible approach where these algorithms are organised in modular units which can be easily plugged in and out of the image evaluation process. These modules are selected, ordered, and configured through a human-readable and easy-to-maintain configuration file.

3.2.2 *Organised image processing with FIRE*

The basic structure of most image processing solutions in physics is the same:

(i) Load image: e. g. read an image file or capture a new image with a camera.

(ii) Preprocess image: cropping, colour conversion, background removal, etc.

(iii) Process image: use computer vision algorithms to extract the desired information from the preprocessed image.

(iv) Export data: save the extracted information to a file or database.

FIRE implements this workflow in a flexible script language. In a configuration file, the user supplies a list of modules for each of the steps above. For each of the files to be processed, FIRE then goes through the following steps (cf. fig. 6):

(i) Based on file type, select the appropriate image loading module, then give the file name (or hardware resource such as a camera)

to the image loader, which returns a byte array containing the image data.

(ii) Hand the image to the first preprocessor, which returns a modified image. This image is given to the next preprocessor, and so forth, until the chain of preprocessing modules is exhausted.

(iii) Give the (fully preprocessed) image to the chain of processing modules. Each processor receives the image and the *detection infos* (e.g. coordinates of a circle or the area of an oil droplet) extracted by processors higher up in the chain, and returns a list of detection infos.

(iv) Give the full list of detection infos to every exporter. They return nothing but typically have side effects (like creating an output file).

In sections 3.2.3 to 3.2.6, we will describe the image processing and computer vision methods used in evaluating images from the experiment described in chapter 2.

3.2.3 *Cleaning up the mess*

Extracting valuable information from raw camera images is (a) hard and (b) computationally expensive. There is no way around both of these properties, but there are ways to reduce the pain they induce. Typical measures to reduce the impact of (a) are background removal, noise reduction, or contrast enhancement. The effects of (b) can often be alleviated by throwing away unusable or redundant parts of an image, such as uninteresting regions or colour information. This section deals with the *pre*processors that were used on this thesis's experiment images to tackle these tasks.

In fig. 7, we see a typical raw image from an experiment, exactly as saved to the SD card by the camera. We make the following observations:

(i) Since the experiment is square, but the camera's CMOS sensor has a 3:2 aspect ratio, roughly a third of the image contains only the dark room that the experiment was set up in.

(ii) The illumination is bright in the center and falls of drastically towards the outside, such that the far corners of the experiment are barely illuminated.

(iii) The background is not noise-free. In this image, wrinkles in the paper screen are clearly visible in the lower two corners.

(iv) There is little information in the colours of the image: while many larger oil droplets have a green-yellowish tint, those towards the center of the experiment, and particularly smaller ones, do not.

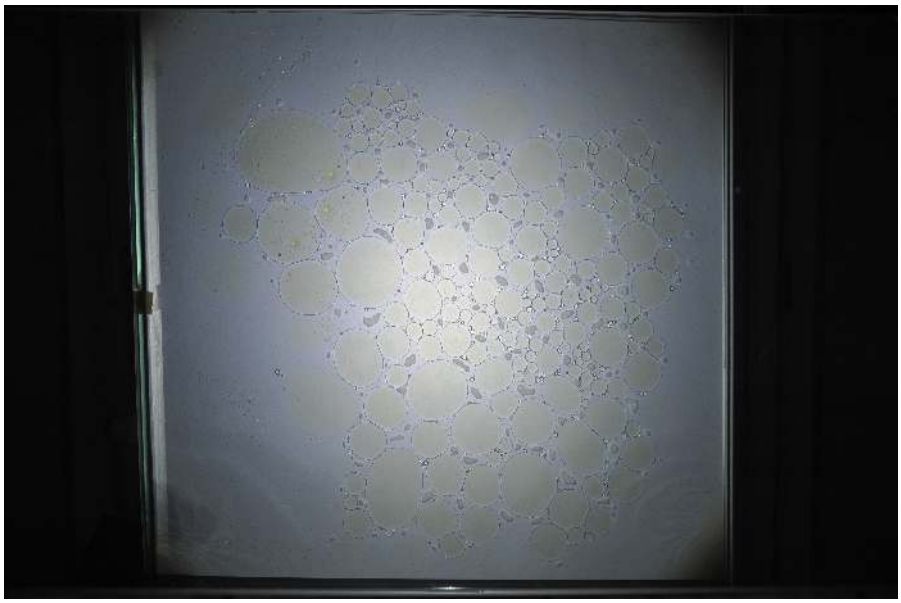


Figure 7: Typical experiment image before preprocessing. Due to its shape, the experiment only occupies roughly two thirds of the image. The background, inhomogeneously illuminated and with noise at the lower two corners, is far from optimal. Little information can be gained from the colour channels, as smaller droplets in general, and larger droplets in the center, do not possess a green-yellowish tint.

To cut down on computing time, we can make use of observations (i) and (iv). First, we crop the image (original resolution 4288×2848 pixels) to the part that contains the actual experiment, resulting in an effective image resolution of 2370×2370 pixels. Second, we reduce the information to be processed by two thirds by converting the colour image to grey scale. A common strategy for this is colorimetric conversion: the information in the three colour channels R, G, and B (for red, green, and blue, respectively) is reduced to a single luminance channel Y via

$$Y = 0.2126R + 0.7152G + 0.0722B,$$

where the three coefficients represent the empirically measured human intensity perception for the colours. Another common method is to throw away two of the three colour channels. For our experiment images, the best contrast at the edges of the droplets is found when using just the red channel. The latter grey-scaling method is used in the following for our image processing.

Figure 8a shows the cropped grey-scale version of our experiment image (fig. 7). The luminosity along a horizontal line shown in light blue is provided in fig. 8b. At the edges of the droplets, where the curvature bends the light towards the inside of the droplet, we find strong negative peaks in light intensity. These will be our main means for the automatic detection of droplets described in the later sections

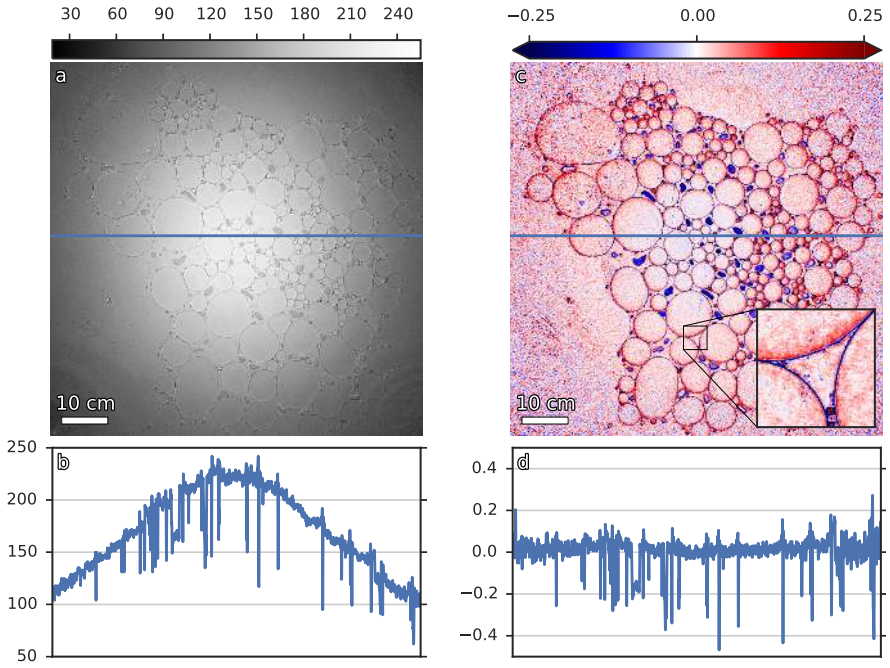


Figure 8: Experiment image cropping and background removal. *a)* Cropped and grey-scaled version of fig. 7. *b)* Droplet edges manifest as negative peaks in the profile along the blue line, but the bell-shaped function overlaid by the background illumination forbids defining a threshold. *c)* Image after dividing by a background reference image (i. e. an image taken just before oil was poured into the setup) and subtracting one. False colours were added to visually separate negative from positive pixels. *d)* We can now separate droplet edges from background via a global threshold. By comparing the signal-to-noise ratios in both profiles it was verified that the perceived increase in noise mainly stems from the different colour maps.

of this chapter. However, the inhomogeneous illumination introduces two difficulties:

(i) A bell-shaped background function is superimposed onto the comb-like function containing the peaks. Consequentially, we can not simply define a threshold to separate droplet edges from the background.

(ii) Not only is the absolute value of a peak dependent on the droplet's position, but so is its difference to the background function. I. e. the height of the peaks with relation to the background decreases with the distance from the image center. Thus, if we subtract the background from the image, we find the peaks enclosed in an envelope function.

As a counter measure to these two difficulties, and to the presence of background noise, we *divide* our image data by the background, where we define the background as a picture taken just before any oil was put into the setup. Furthermore, we subtract one, such that a negative pixel value corresponds to "darker than background", with -1 marking absolute darkness. Similarly, a positive pixel value corresponds to "brighter than background", with +1 marking a doubling in brightness. Note that the latter step is equivalent to subtracting the background from the image before dividing by it.

The background-adjusted image is shown in in fig. 8c, with false colours to separate positive and negative values. In comparison to the original image on the left, we see an increase in contrast between droplet edges and background near the edges of the system. Additionally, the wrinkles at the bottom corners of the image disappeared (while there are no droplets in these corners in this image, they are common at earlier times during an experiment). A look at the profile (fig. 8d) reveals that the bell-shaped background was almost fully removed, and the position dependence of the peak height is strongly suppressed. Droplet edges are now much better separated from the background. In passing we note that the perceived increase in noise in the processed image is mostly a result of the tighter colour map: by inspection of the two profiles, we see that the signal-to-noise ratio is invariant, or at most marginally increased, upon background removal.

3.2.4 *Small oil droplets are circles!*

In computer vision, it is crucial to know characteristics of the objects one wishes to detect. That is, it is much easier to identify an elephant in an image when you *expect* that there is an elephant in the image. In fig. 7, we see that smaller oil droplets are almost perfectly circular, while the largest droplets have less well-defined forms, ranging from ellipsoidal to irregularly, i. e. potato-shaped. This section deals

with identifying the small droplets by making use of their circular characteristic. The detection of larger droplets will be addressed in section 3.2.5.

As an introduction, I describe the top dog of circle detection algorithms: the Circle Hough Transform;[64] its advantages and shortfalls, and why it is inapplicable for our purpose. Next, we explain a novel, fast approach by Rad et al.[65] that was implemented in Python and C as part of the present thesis, and that generates a list of circle candidates from an image as shown in fig. 8c. Subsequentially, we cluster these circle candidates to retrieve a list of detected circles via a well-established density-based clustering algorithm.[66] In a final step, we eliminate an easily identifiable source of experimental noise by making use of directional statistics.

3.2.4.1 *Conventional circle detection*

Due to their abundance in nature as well their simple mathematical properties, circle detection, or, more generally, the detection of curved objects, is one of the oldest fields in computer vision (cf. Jain et al.[56], ch. 6 or Illingworth and Kittler[67]). Consequentially, there are many well-established algorithms to detect circles. The most prominent of these is the *Circle Hough Transform* (CHT), a specialisation of the Hough Transform invented by Duda and Hart[64], based on a 1962 patent by Hough.[68] The rough idea behind CHT is as follows (colours refer to fig. 9):

(i) Assume that we want to detect circles with known radius R.

(ii) Detect edges in the image, such that dark pixels correspond to object edges and bright pixels correspond to uninteresting background. In other circumstances, a popular method to find edge points is Canny Edge Detection[69]. In fig. 8c, the droplet edges already correspond to negative intensity peaks, and we can arrive at a black-and-white image through thresholding.

(iii) Around every dark (edge) pixel, draw a circle with the known radius R (light blue in fig. 9).

(iv) Split the image into equally-sized bins (the *accumulator matrix*). Every bin that is touched by a drawn circle receives a *vote* from this circle. After evaluating all drawn circles, we report a detected circle with radius R at the center of each bin where the count of votes is a local maximum and exceeds a suitably defined threshold value (dark red star in the center of fig. 9).

There are many modifications[70] to the Circle Hough Transform which make it faster, consume less memory, or more robust to noise. Still, most of these suffer from a basic drawback of CHT: if the circle

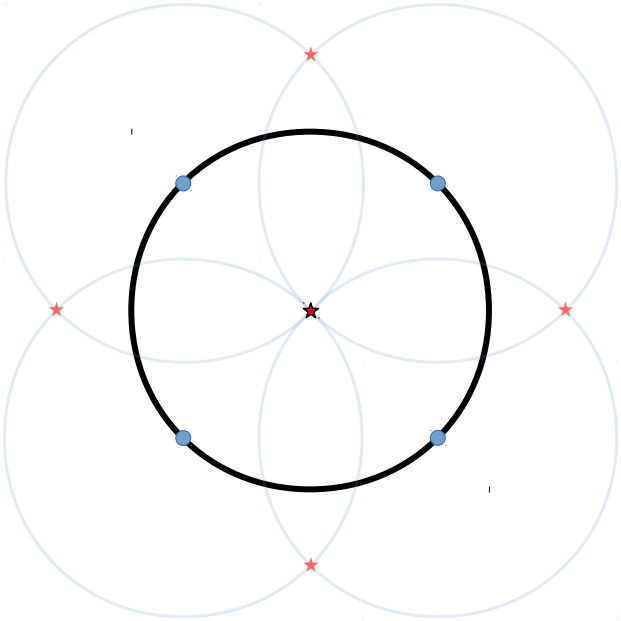


Figure 9: Circle Hough Transform. Draw a circle (light blue) with radius R around every edge pixel in the original image (black, done exemplarily at dark blue points). Points where the drawn circles overlap (red stars) are candidates for a circle with radius R in the original image. The candidates are binned in the three-dimensional circle parameter space, and local maxima in this space (above a threshold) are considered "real" circles (dark red star in center). The algorithm should be repeated for a range of R when the circle radius is unknown.

radius R is unknown, the algorithm must to be repeated for each radius of interest. If the range of possible radii is large, like in our oil droplet experiment, this becomes very computationally intensive, even on modern computers.

3.2.4.2 *Improving robustness and efficiency with FCD*

In 2003, Rad et al.[65] proposed *Fast Circle Detection* (FCD): a novel, size-invariant method for circle detection. Averaged over a variety of sample images from different application fields, they report not only an improved robustness against noise, but also execution times more than thousand times faster than these for CHT, and more than eighty times faster than a popular improvement on it, the edge-oriented CHT.[71] Although Rad et al.[65] originally branded FCD as only working on "circles which are totally brighter or darker than their backgrounds",[65] we successfully apply it to our "edge-like" greyscale images (see (ii) above) without modifying the underlying idea.

The basic concept of FCD is to make use of the circle's symmetry by finding opposing gradient vectors. Consider fig. 10a: Along the circle, we find strong gradients pointing towards its center. For each of the gradient vectors, we find a complementary one at the opposite side of the circle. Each of the gradient vector pairs is translated into a *circle candidate* (x, y, r), where x and y mark the center between the two gradient vectors, and r marks their halved distance. I. e. (x, y, r) corresponds to a circle with radius r that touches the bases of both gradient vectors. FCD has two parameters to accommodate that circles in "real" images are typically not perfect. These parameters dictate which gradients are considered pairs (cf. fig. 10b):

(i) The gradient vectors of a perfect pair point at exactly opposing directions, i. e. their angles differ by exactly 180°. The parameter α determines a tolerance around this difference.

(ii) The line that connects the bases of a perfect gradient vector pair is aligned with the directions of the two vectors, i. e. its angle matches the direction of one vector. The parameter β determines a tolerance around this direction.

A naïve implementation could compare every gradient vector to every other gradient vector, such that the computational effort scales as $\mathcal{O}(n^2)$, where n is the number of pixels, and hence the number of gradient vectors, in the image. As our images are very large (2370 × 2370 pixels after preprocessing), we take some simple measurements to decrease the computational burden:

(i) Only gradient vectors with a magnitude larger than a *minimum norm* are considered. This *replaces* the thresholding step and rules out most gradient vectors that lie in the background and inside oil droplets.

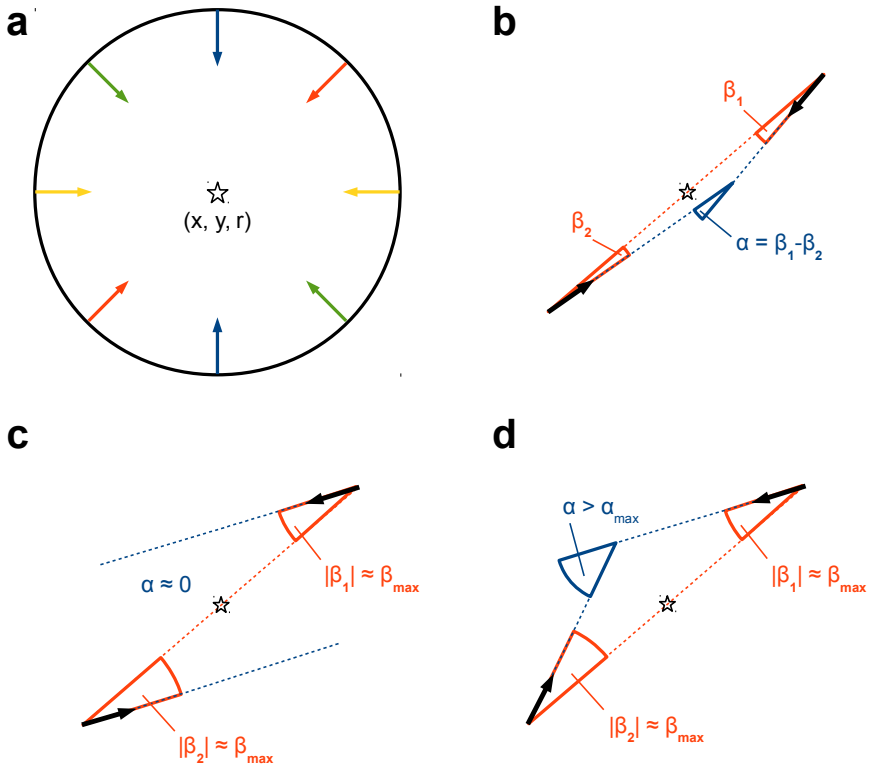


Figure 10: Fast Circle Detection (FCD). *a)* In an edge image, there are strong gradients pointing towards the center of a circle. FCD looks for pairs of opposing gradient vectors (colour-coded), and translates them into circle candidates (black star, one candidate per vector pair). *b)* FCD has two parameters that determine which gradient vectors are considered a pair. The allowed deviation in vector direction is determined by α, while β sets the allowed deviation between vector direction and the direction of the line connecting them. *c) and d)* α filters circle candidates originating from non-parallel vector pairs. While the star in c) could be the middle of a noisy small circle, the star in d) would be placed more appropriately at the intersection of the two dotted blue lines.

(ii) A third filter parameter, γ, is introduced. A possible vector pair is no longer processed when the ratio of their magnitudes is larger than γ.

(iii) The vectors are stored in memory not ordered by position, but ordered by their angle.

Hence, the implementation written for and used in this thesis adopts the following workflow:

(i) Calculate the gradient vector for every pixel of the image using the NumPy library[72]

(ii) Drop all gradient vectors that have a magnitude smaller than the *minimum norm*, then sort the remaining ones by their angle

(iii) For all vectors (until break condition):
   a) Given a vector with angle φ, find the first vector with an angle larger than $\vartheta = \varphi + \pi - \alpha$. If $\varphi > \pi$, break out of loop
   b) For all vectors with angle larger than ϑ (until break condition):
      i. If angle of this vector is larger than $\varphi + \pi + \alpha$, break out of loop
      ii. If ratio of norms is larger than γ, continue with next potential partner vector
      iii. Calculate angle of line connecting the two vectors (this is by far the most expensive step)
      iv. If difference between the angle just calculated and φ is larger than β, continue with next potential partner vector
      v. If all the checks above passed, save a circle candidate, with the radius corresponding to half of the distance between the two vectors, and the center lying midway between them.

In our final image analysis, due to the irregular shapes of big droplets, we only detect circles up to a maximum radius r_{max}, where r_{max} is much smaller than the image dimensions. We use this to significantly reduce the computing time by splitting the image into smaller tiles that have an overlap of $2.2 \cdot r_{max}$, and performing FCD on each of the tiles separately.

3.2.4.3 *Subpixel-resolution clustering*

After these steps, we are left with a long list of circle candidates in a three-dimensional circle parameter (x, y, r) space. Just like in the

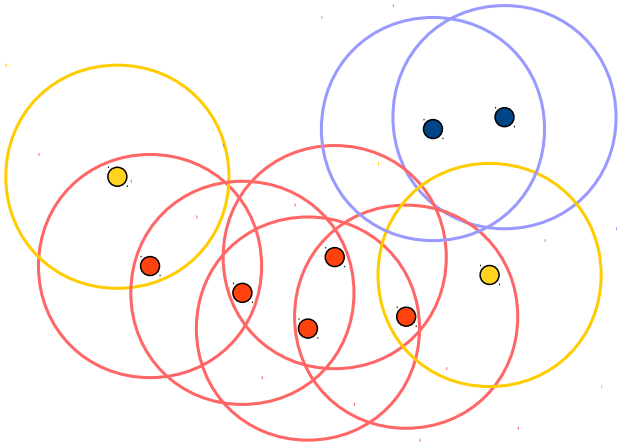


Figure 11: DBSCAN with `minPts` $= 2$. If a point has at least two other points in its ϵ environment, it is considered a *core point* (red). All points that are reachable from a core point, or reachable from the points in its environment (and so on), are considered part of the same cluster (red and yellow). Points that have less than two other points in their ϵ environment, but are reachable from a core point, are considered *edge points* (yellow), and are part of the same cluster as the core point(s) they are reachable from. Per definition, no point is reachable from an edge point. If a point has less than two other points in its ϵ environment and is not reachable from a core point, it is considered *noise* (blue).

original CHT paper[64], we could generate a list of circles from these by binning and finding local maxima above a threshold. This is the method used by Rad et al.[65]. However, binning comes with a list of well-known drawbacks, such as loss of resolution or a strong result dependence on bin size; and not only has the world of circle detection evolved since the 1970s, but so has the world of cluster analysis, i. e. the world of segmenting a heterogeneous population into a number of more homogeneous subgroups[73] – in our case, the decision which circle candidates belong to a circle in the original image, and which candidates are regarded as noise.

A modern and particularly well-established (cf. Berkhin[74], p. 43) clustering algorithm is *DBSCAN* (*density-based spatial clustering of applications with noise*, proposed by Ester et al.[66] in 1996. It is a density-based clustering algorithm: For a given a set of points, it groups together those points that are closely packed together, and marks single points in low-density regions as outliers (noise). This makes it well suited for our FCD data, where circle candidates are the points in the (x, y, r) circle parameter space, and clusters of points that are similar in all three parameters (i. e. close neighbours in the parameter space) are likely candidates for a circle in the original image.

DBSCAN classifies the points in some space into three classes (cf. fig. 11):

(i) A point p is considered a *core point* if at least `minPts` other points are within distance ϵ of it. These other points are considered *directly reachable* from p.

(ii) A point is considered an *edge point* if it is directly reachable from a core point, but has less than `minPts` points in its ϵ environment. Per definition, no points are directly reachable from an edge point.

(iii) A point is considered *noise* if it is *not* directly reachable from a core point and has less than `minPts` points in its ϵ environment.

A point q is considered *reachable* from a point p if there is a path $p_1, \cdots, p_n$, with $p_1 = p$ and $p_n = q$, such that p_{i+1} is always directly reachable from p_i. If a point is a core point, it forms a *cluster* with all points that are reachable from it. After the classification above, we end up with a set of distinct clusters and a set of single outlier points (noise). Each cluster contains at least one, possibly more, core points, and possibly a number of edge points.

The classification depends on only two parameters: `minPts` and ϵ. It is performed as follows:

(i) Pick a random point that has not been visited.

(ii) If the point is *not* are core point (i. e. if there are less than `minPts` points within distance ϵ of it), mark it as noise and go back to (i). The point will possibly be detected as an edge point later in the process.

(iii) If the point *is* a core point, start a cluster containing it and all the points in its ϵ environment.

(iv) Visit all the points in the cluster, and, if they are also core points, add the points in their ϵ environments to the cluster. This process continues recursively.

(v) When the complete cluster has been visited, go back to (i). If there are no further unvisited points, our work is done.

In our image analysis we use an implementation of DBSCAN provided by scikit-learn.[62] The clusters found by DBSCAN correspond to circles in the original image, and we report one detected circle per cluster, at the (x, y, r) center of mass of its circle candidates.

3.2.4.4 *Reducing false detections*

Circle detection often aims at finding a single, or few, circles in an image, e.g. in iris recognition or fingertip positioning. However, in our case, the images contain thousands of closely packed circles. This introduces clusters of false circle candidates between two circles (cf. fig. 12). These clusters differ from the ones that result from real circles

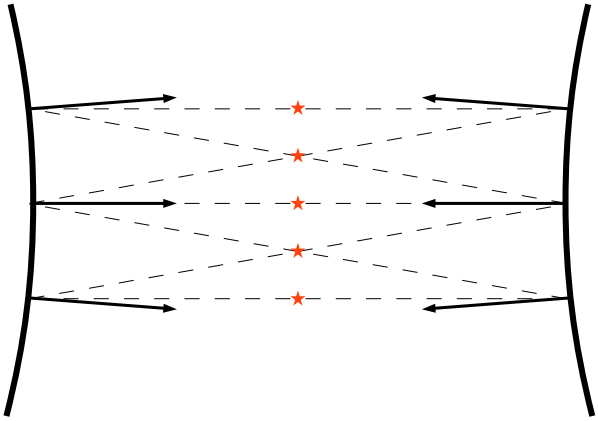


Figure 12: False circle candidates. Since FCD accepts slightly non-perfect gradient vector pairs (managed through the α and β parameters, cf. fig. 10b), it generates false positives (red stars) between two circles lying next to each other. The resulting clusters of circle candidates can be filtered by requiring the directions of the connecting vectors (dashes lines) to be uniformly distributed.

(e.g. fig. 10a): The angles of the lines connecting the gradient vector pairs are not uniformly distributed, but restricted to a narrow range.

To remove the false positives we now wish to quantify the uniformness of our angular data. First, we identify that the lines connecting the vector pairs are *bidirectional*: None of the two gradient vectors is special, and a connecting line with angle φ is identical to a line with angle $\varphi \pm \pi$. Hence, our angular data effectively only lives in the $[0, \pi)$ range, not in a $[0, 2\pi)$ range of directional data. To use principles from the more common unidirectional circular statistics (mainly revolving around the trigonometric functions, see e. g. Fisher[75], ch. 2 for an introduction), we transform our bidirectional angles (said to live in a *diametrically bimodal circular distribution*) to the full $[0, 2\pi)$ angle space by *angle doubling*[75], i. e.

$$\varphi \to \vartheta = 2\varphi \mod 2\pi.$$

We can then quantify the angular spread of our angles $\vartheta_i, \cdots, \vartheta_N$ via the measure[75]

$$\begin{aligned} r &= \sqrt{X^2 + Y^2}, \text{ with} \\ X &= \frac{1}{N} \sum_{i=1}^{N} \sin \vartheta_i, \\ Y &= \frac{1}{N} \sum_{i=1}^{N} \cos \vartheta_i. \end{aligned}$$

As the sine and cosine have vanishing integrals over their periods, r vanishes for perfectly uniform angle distributions, and it approaches

unity when the distribution prefers a particular angle. Hence, to separate noise clusters as in fig. 12 from clusters belonging to real circles, we stipulate a maximum angular spread r_{max} for the set of connection angles belonging to the circle candidates of a cluster. In our image processing, I found $r_{max} = 0.6$ a good compromise to filter clusters lying between two circles (as in fig. 12) while still allowing clusters arising from small circles which have few circle candidates (and hence possibly a narrow angle distribution due to noise).

3.2.5 *Larger droplets are potatoes!*

In our experiment images (figs. 7 and 8), we can easily identify that large droplets do not always come in a circular shape. As we will see in section 4.1.1, this arises from the fact that, after a merging event, large droplets require much longer times to relax into a circle shape than small droplets. Hence, the oil droplet characteristic we rested upon to detect small droplets is not reliable for large droplets. This section deals with detecting the potato-shaped larger droplets by thresholding the image to find the dark droplet edges, then performing morphological operations and extensive filtering. This approach, while working, is much less sophisticated than the circle detection approach, and developing it relied heavily on a trial-and-error workflow.

3.2.5.1 *Segmenting the image into connected areas*

In fig. 8, we can identify a property that is common to all droplets, and in particular, that is independent of their shape: At its edge, the droplet's curvature bends the incident light, forming a dark outline of the droplet. Using this outline to split the image into a set of areas is the heart of this and the following section. Subsequently, section 3.2.5.3 deals with identifying which of these areas are considered oil droplets.

A popular method for image segmentation is the *watershed transform*. Its inspiration stems from geoscience, and Vincent and Soille[76] give a perspicuous and to-the-point introduction that I wish to reproduce and not rephrase:

> Everybody has heard for example about the *great divide*, this particular line which separates the U.S.A. into two regions. A drop of water falling on one side of this line flows down until it reaches the Atlantic Ocean, whereas a drop falling on the other side flows down to the Pacific Ocean. As we shall see in further detail later, this great divide constitutes a typical example of a *watershed line*. The two regions it separates are called the *catchment basins* of the Atlantic and the Pacific Oceans, respectively. The two

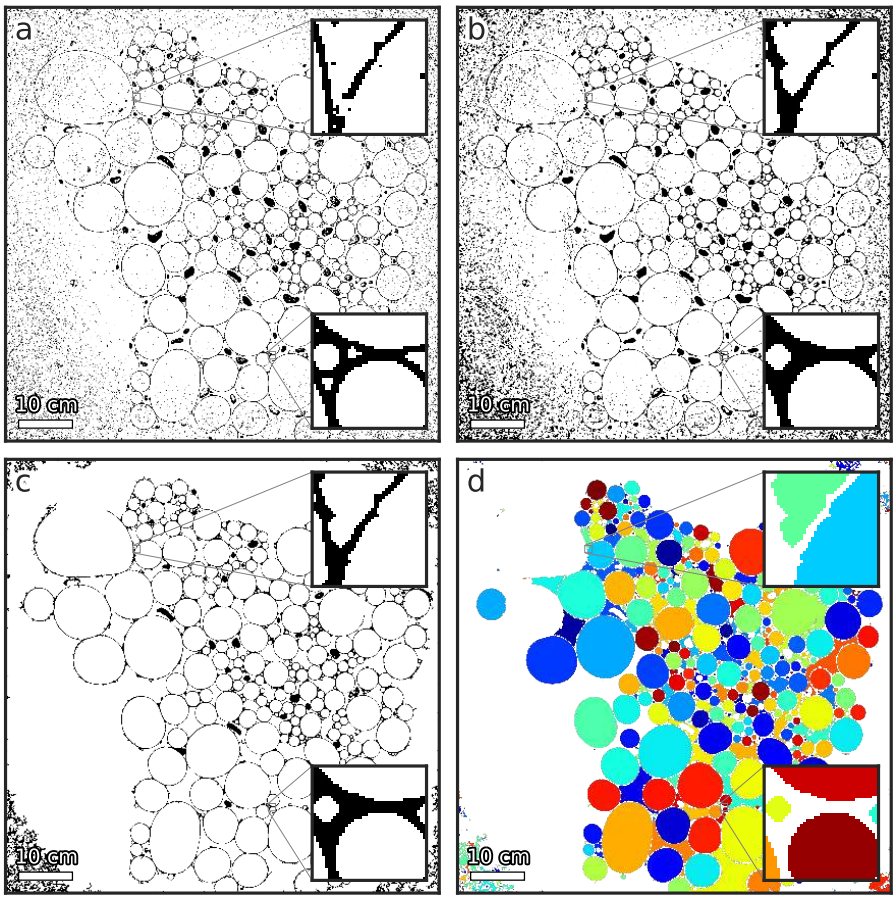


Figure 13: Detection of large droplets. *a)* Through thresholding the preprocessed image (fig. 8c), we can do a rough identification of the droplet edges. However, the image is so noisy that gaps emerge in the edges (upper inset), as do sprinkled dark spots that do not belong to a droplet outline. *b)* Via *erosion* and *dilation*, we can close many of the gaps in the droplet outlines. *c)* Through *binary filling*, i. e. through filling dark regions that do not contain any white pixels, we remove the sprinkled dark spots. *d)* Labeling the connected white regions yields many candidates for droplets, but also quite a few false detections and missed droplets.

> Oceans are the *minima* associated with these catchment basins.

The idea of the watershed transform is to interpret our experimental images as a height map, where the light intensity at a pixel corresponds to the height at that pixel. For every pixel of the image, we follow the gradient until we reach a local minimum. All pixels from which we end up at this minimum, i. e. all pixels that are in this minimum's catchment basin, receive the same label (e. g. a number). These labels designate the areas into which we segmented our image. Note that, since our droplets are bright in the middle and dark at the edges, we would invert the image before applying the watershed transform.

In our case, the watershed transform struggles with two issues, ultimately making us refrain from using it:

(i) There is a high level of noise (cf. fig. 8c), particularly towards the borders, introducing many more regional extrema than there are droplets*. This leads to *over-segmentation*, where single oil droplets are segmented into multiple areas.

(ii) The droplets sometimes have small gaps in their dark outlines, an artifact of either the imaging or the droplets position relative to other droplets, making the detected areas "leak out" at these gaps.

The approach to finding large droplets adopted for the present work is based on detecting the droplet outlines as a binary (black-and-white) image, then filling them (much like the fill tool you remember from *Microsoft Paint*) to find the droplet areas. But even when not using the watershed transform, the issues above persist.

There are several ways to convert a grey scale to a black-and-white (i. e. binary) image. The most simple one, and the one we use, is to simply *threshold* the image. For a given threshold grey value t, we map the value of all pixels with an original grey value larger than t to 1, and the value of all other pixels to 0. While doing this, we again encounter the two issues: If we choose t too negative, many of the droplet outlines contain grey values above the threshold, which introduces gaps in the outlines. On the flip side, the closer we move t towards zero, the more sprinkled black dots (arising from noise, cf. fig. 8b) will end up in the thresholded image. Figure 13a shows our preprocessed sample image after thresholding with $t = -0.05$. The threshold is a compromise between the two extremes, such that we find only few gaps in the droplet outlines and a medium level of

* The typical cure for noisy images is to blur them in preprocessing, e. g. with a Gaussian filter. However, with our images, a reduction of the noise to acceptable levels required such heavy blurring that the droplets themselves were strongly smeared out.

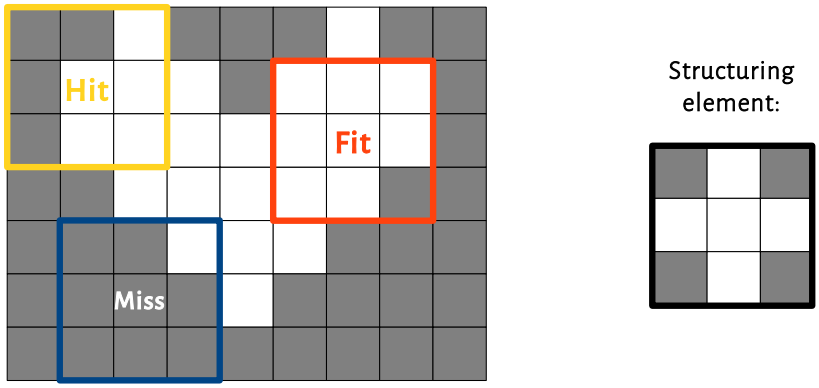


Figure 14: Morphological testing. The image is probed with a *structuring element*, in this case a 3×3 cross (right). The structuring element is positioned at all possible locations and compared with the corresponding pixel neighbourhood. Yellow: The structuring element hits (intersects) the neighbourhood. Red: The structuring element fits the neighbourhood. Blue: The structuring element misses the neighbourhood.

noise. In the next section, we work towards solving both of these issues through *mathematical morphology*.

3.2.5.2 *Morphology*

Mathematical morphology is one of the oldest techniques in digital image processing. It originates from the study of porous media by Matheron[77] and Klein and Serra,[78] who considered points on a grid representing a binary state, i. e. either porous or non-porous. In short, mathematical morphology is a theory for the analysis of spatial structures,[79] with *mathematical* referring to its basis in set theory, integral geometry and lattice algebra, and *morphology* referring to its aim of analysing the shape and form of objects. Morphological operations probe an image with a small template called the *structuring element*. This template is positioned at all possible locations in the original image and compared with the corresponding pixel neighbourhood (cf. fig. 14). The structuring element is said to

- *fit* the pixel neighbourhood when for every pixel of it that is set to 1 the corresponding pixel in the neighbourhood is also set to 1,
- *hit* (intersect) the pixel neighbourhood when for *at least one* pixel of it that is set to 1 the corresponding pixel in the neighbourhood is also set to 1, and
- *miss* the pixel neighbourhood if none of the above cases is true.

The result of a morphological operation is a new binary image, where every pixel corresponds to the result (success or failure) of the test at

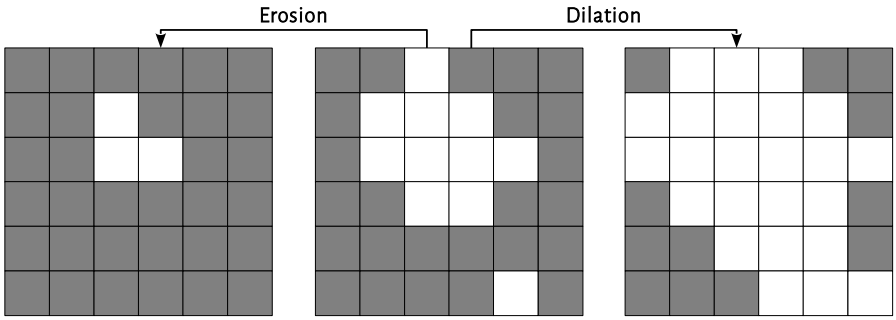


Figure 15: Erosion and Dilation with a 3×3 cross. Erosion answers whether a pixel neighbourhood fits the structuring element (cf. fig. 14), and intuitively corresponds to making white (nonzero) areas shrink. Dilation has the opposite effect: Answering whether a pixel neighbourhood intersects the structuring element, it makes white areas grow.

that position. A deeper introduction is way beyond of the scope of this thesis, but I encourage the reader to take a look at one of the definitive books in the field, written by Pierre Soille.[79]

The morphological operations we are particularly interested in are its two workhorses for noise removal, *erosion* and *dilation*:

- *Erosion* is the answer to the test "*Does the structuring element fit the neighbourhood?*" For typical structuring elements, e. g. 3×3 crosses (see fig. 14) or 3×3 blocks, this corresponds to the intuitive sense of geological erosion: blocks of white (i. e. nonzero) pixels lose elements at their edges, as these neighbour black pixels and therefore do not fit the structuring element (cf. fig. 15).

- *Dilation* has the opposite effect: it answer the test "*Does the structuring element hit the neighbourhood?*", and increases the size of blocks of white pixels as the answer is "yes" for a black pixel adjacent to at least one white pixel (again assuming a 3×3 cross as structuring element).

In our thresholded experiment images, we can close the gaps in the droplet outlines by performing erosions (i. e. intuitively, by making the black areas grow). When we perform the same number of dilations afterwards (to approximately restore the original area of the droplets), the gaps will not be re-opened, since they are surrounded by other dark pixels from the outline. This allows us to (partly) solve issue (ii): Figure 13b shows our thresholded sample image after performing two erosions followed by two dilations. The closing of an outline gap can be seen in the upper inset.

After eroding and dilating, we (ideally) have an image where every droplet has an intact outline. Using this characteristic, we can deal with issue (i): By converting all black clusters that do *not* embed a

white cluster (as our droplet outlines do), we almost fully remove the sprinkled noise. The result of this operation can be seen in fig. 13c, where we find remaining noise only at the corners of the image. This arises from both the high level of noise in these regions and the fact that the hole-filling algorithm as implemented SciPy[80] does not convert black clusters that touch the image border, even when they do not embed a white cluster. At this point (ideally), our droplets are unbroken white clusters that are separated from the rest of the image through an embedding black cluster.

We now identify candidates for individual droplets by assigning labels (e. g. numbers) to the white clusters. The result of this (with colour-coded labels) is shown in fig. 13d. While we were able to correctly segment many of the droplets from our original image, there are two major issues:

(i) There are many false detections in the space between droplets.

(ii) There are a few missed droplets, in particular large ones. This partially arises from different levels of light refraction depending on the droplet size, with larger droplets having less well defined outlines.

We deal with issue (i) in the next section, separating the false detections from real droplets through extensive filtering. In the final image processing, issue (ii) is overcome by using multiple parameter sets for our non-FCD droplet detection, i. e. by using different threshold values, erosion and dilation steps, and filters (cf. section 3.2.6).

3.2.5.3 *Meet the filters*

In fig. 13d, we find the results of our image segmentation through thresholding and morphological operations. They are not exemplary for our final image processing (cf. section 4.1.4), where we use more finely tuned parameters and a combination of parameter sets for different droplet sizes. Rather, they show a fundamental challenge for our image segmentation: The droplets are packed so closely that their outlines often touch each other. This creates isolated white clusters that do not correspond to droplets, but to the space in between them. However, our labeling has no method to separate between these two types. Hence, we find many *false positives* in our segmented image.

At the beginning of section 3.2.5 on the detection of large droplets, we ruled out using Fast Circle Detection as the droplets do not always possess a circular shape. Still, there are other characteristics which we can use to divide droplets and false positives. In this section, we introduce simple filters, aimed at the characteristics of our droplets, to separate real droplets from detections corresponding to the in-between spaces. After preprocessing, thresholding, morphing, filling, and labeling the image, we run all detected areas, separately,

through a chain of filters. The filters each calculate a certain property of the area, then check whether it fits into a range of acceptable values.

First, we calculate the *size* of the area by counting the pixels that it is comprised of. While this bears no potential to separate false positives from oil droplets, it is useful for two things: First, we use it to deal with issue (ii) from the end of section 3.2.5.2, i. e. the fact that very large droplets have thinner outlines. By running different parameters, e. g. more erosion steps, for just these droplets, we are able to close large gaps in their outlines without sacrificing precision in our detection for smaller droplets. Second, we use it to separate the realm of thresholding-and-morphing detection from the realm of the more precise Fast Circle Detection.

Next, we calculate the *standard deviation* of the original (preprocessed) grey values of all pixels that belong to the inspected area. This makes use of the fact that droplets never live inside other droplets. The erosion and filling operations will sometimes completely eliminate small droplets from the binary image (as the erosion converts any white pixels on the inside to black, and filling then removes the all-black "hole" from the image). The areas that originally contained these small droplets will then have a large standard deviation in pixel grey values, as the small droplets originally had very dark edges and very bright centers. Since oil droplets do not live inside other oil droplets, high standard deviations are only observed in false positives. This filter fails, however, for false positives that contain no droplets, e. g. in areas near the edge of the experiment.

The third filter calculates the *total corner response* of the area. Since the shape of the oil droplets in mainly governed by interface tensions, they almost always possess a smooth outline. On the other hand, false positives, circumscribed by the outlines of multiple droplets, often have sharp corners at the points where the oil droplet outlines adjoin. Corner detection, being very useful for finding points of interest, is a wide field in computer vision.[81,82] Here, we use one of the most popular corner detection algorithms, that is due to Harris and Stephens,[83] and implemented in the scikit-image[61] library. The basic idea, going back to an original approach by Moravec,[84] is very simple: Given a pixel in the image, we consider a patch of the *gradient image* centered on this pixel, and compare it to nearby, largely overlapping patches (cf. fig. 16) by considering the sum of squared distances (SSD).

(i) When the pixel is in a mostly uniform area, the contents of the patch will change little, regardless of the direction in which it is moved.

(ii) When the patch contains an edge, its contents will show a large variation when moved perpendicular to the edge, but little variation when moved along the edge.

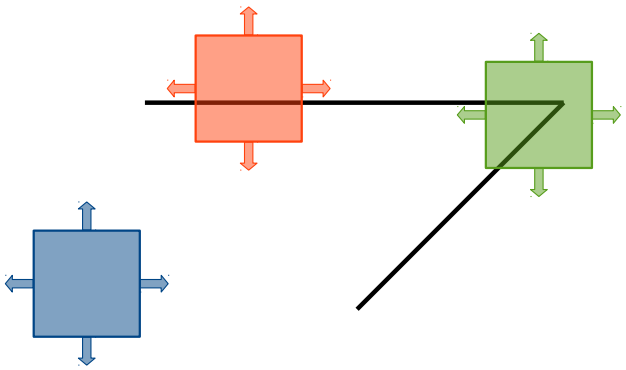


Figure 16: Corner detection after Moravec.[84] A patch around a center pixel is compared to nearby, largely overlapping patches by considering the sum of squared distances. The smallest sum of squared distances over all patch movements is the corner response of the considered pixel. *Blue*: Pixel in a mostly uniform area, where the contents of the patch change little when moved around (small corner response). *Red*: Pixel near an edge, where the contents of the patch change a lot when moved perpendicular to the edge, but little when moved along it (small corner response). *Green*: Pixel near a corner, where the contents of the patch change a lot regardless of the direction in which it is moved (large corner response).

(iii) When the patch contains a corner, i. e. an intersection of two edges, it will show a large variation regardless in which direction it is moved.

The *corner response* of our pixel is the smallest SSD between the patch centered on our pixel and any of its neighbours. A large problem with Moravec's approach, as pointed out by himself, is that it is not isotropic: If an edge's direction does not match the direction of the image axes, the smallest SSD will be large even for case (ii). Harris and Stephens[83] solve this by considering the SSD in all directions through a Taylor expansion. We calculate the total corner response of an inspected area as the sum of corner responses of its individual pixels (calculated after Harris and Stephens[83]), divided by the length of the area's perimeter. This value will be large for areas with many sharp corners and small for areas that have a smooth outline. Hence, we require all areas to have a maximum total corner response TCR_{max}.

While the large droplets are often not perfectly circular, their shape, sometimes elliptical, sometimes circular with bumps, typically still resembles a circle. Using this quality, we can further filter some false positives. Given an area A with perimeter p, the *isoperimetric quotient* Q is the ratio of A and the area that a circle with perimeter p would have,[85] and measures the object's compactness:

$$Q = \frac{4\pi A}{p^2}$$

As circles are the shape that maximise the area under the constraint of a fixed perimeter (i. e. the shape that maximises compactness), $0 \leqslant Q \leqslant 1$, where the second equality holds only when the examined area A is circle-shaped. We can filter some of the false negatives, whose shapes sometimes strongly differ from that of a circle or an ellipse, by requiring all detected areas to have a minimum isoperimetric quotient Q_{min} to be considered a droplet.

Our last filter is similar to the isoperimetric quotient, as it also aims at the fact that most large droplets are shaped similar to circles or ellipses. This makes them (typically) convex. On the other side, as many false positives consist of areas in between the droplets, they are shaped complementary to these droplets, and hence are (typically) piecewise concave. There are many measures for concavity.[86,87] For its ease of calculation and its intuitive meaning, we define the *convex ratio* CR of an area A as:

$$CR = \frac{A}{A_{CH}},$$

where A_{CH} is the area of the *convex hull* around A. The convex hull for a set X (in our case the pixels that make up A) is the smallest convex set that fully contains X. Similar to the isoperimetric quotient, if A is already convex, we would find $CR = 1$, whereas we find CR increasingly closer to zero the more concave A is. As with the isoperimetric quotient, we stipulate a minimum CR for a detected area to be considered a droplet.

3.2.6 *Putting it all together*

Having a long list of computer vision algorithms, morphological operations, and filters sounds great until you realise that none of these tools work without the parameters appropriate for the problem at hand. There simply is no universal recipe to finding suitable parameters for a specific problem, and more often than not it ends in a few hours of trial-and-error.

The good news is: there are means to ease the trial-and-error process! For the detection of small droplets, i. e. for the Fast Circle Detection and DBSCAN algorithms, I developed a *parameter sandbox* (fig. 18) based on the matplotlib[88] and PyGObject[89] libraries. The sandbox allows changing the many FCD and DBSCAN parameters and provides almost instant feedback on the circle candidates and circles detected in a small image segment. For the larger droplets, FIRE is flexible enough to show the output of some filters, e. g. the corner response or the isoperimetric quotient, before filtering areas outside of the criteria.

The full FIRE configuration file as used in the final image processing is shown in listing 1 (page 83). For detecting large oil droplets,

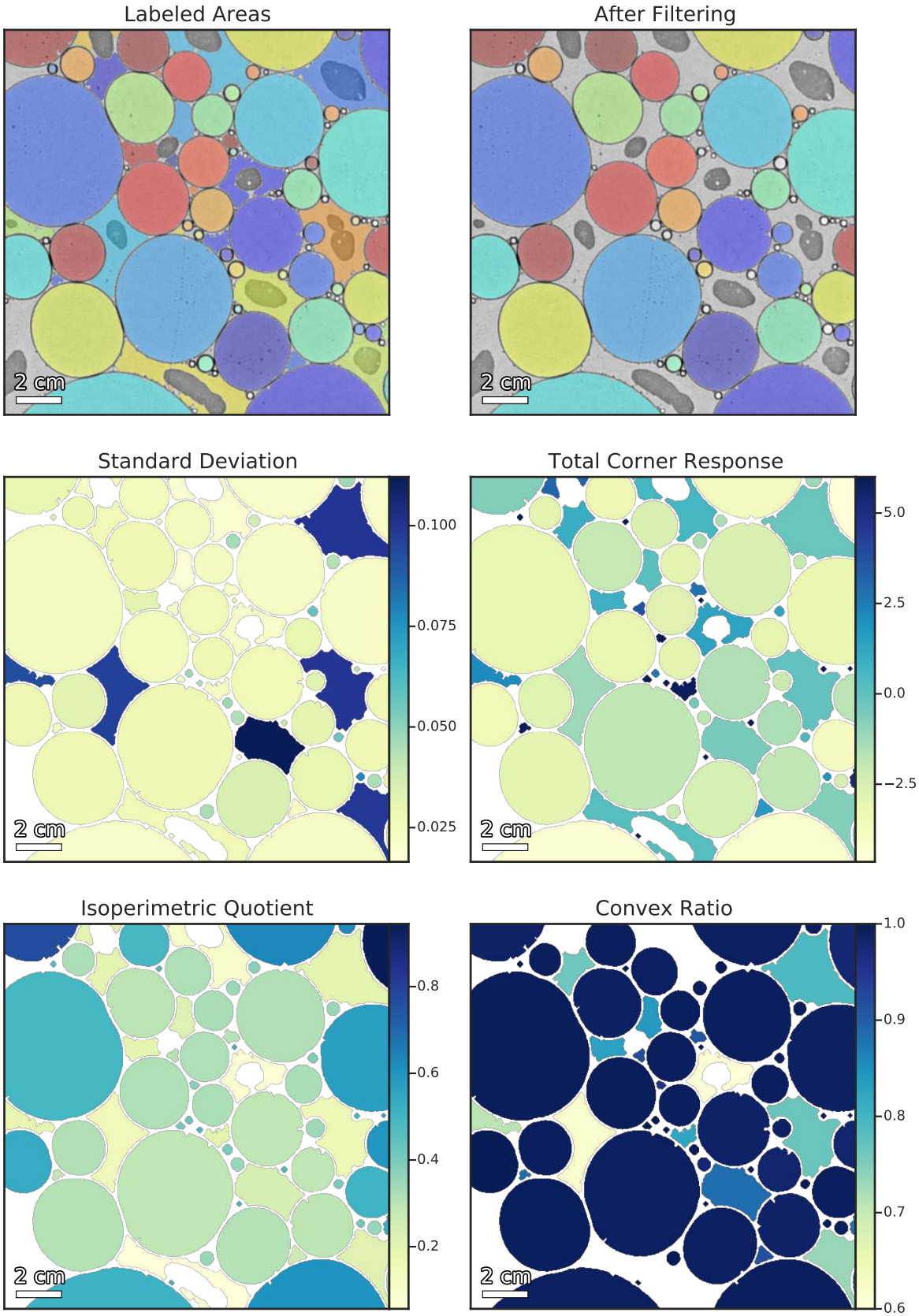


Figure 17: Filter performance for large droplets. Using common characteristics of large droplets, such as their uniform inside, their smooth outlines, their compactness, or their convexity, we can separate them from false positives (i. e. mostly areas in between the droplets). See section 3.2.5.3 for explanations on the four characteristics used.

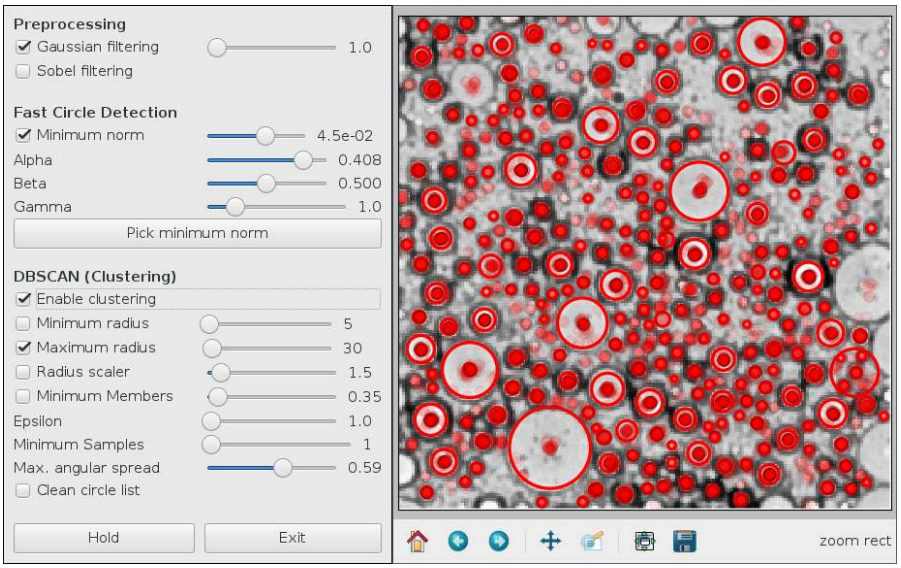


Figure 18: Parameter Sandbox for Fast Circle Detection. The parameters for FCD and DBSCAN entered with the many sliders on the left-hand side are immediately applied to a small portion of a pre-processed experiment image (right-hand side). Circle candidates are overlaid as small red dots (as a projection in which the information on radius is lost). Detected circles gathered from these candidates through clustering are shown as red circles.

we use three different combinations of thresholds, morphological operations, and filters. After each threshold-morph-filter process, the areas of the detected droplets are removed from the image and will no longer be picked up by any further threshold-morph-filter combinations, or by the FCD algorithm. First, we detect very large clusters with many erosions and a relatively weak requirements towards convexity: very large false positives are rare, and when they occur, they typically wrap around the center area that contains droplets, making them very concave. Next, we detect clusters that we are particularly certain about, with only one erosion and strong requirements towards the maximum standard deviation, maximum corner response, minimum isoperimetric quotient and minimum convex ratio. This stage picks up most of the droplets that are outside of the area range we use for FCD. Afterwards, we try to recover missed middle-sized droplets with no erosions, but a threshold that is relatively close to zero, and even stronger critical values for the filters. Finally, we use Fast Circle Detection to detect small droplets down to just few pixels. The parameter set used is quite loose in the restriction of gradient vector pairs, producing many circle candidates, but filters false positives with a stronger requirement towards the angular spread.

4

RESULTS

In chapter 3, we captured images from the experimental setup and made extensive use of computer vision algorithms to process them into lists of detected droplets. With this list, we are finally able gain insight into the physical properties of our coalescing oil droplets on a water surface. In section 4.1, we first take a look at the phenomena that can be seen with the bare eye: In a typical experiment time series, more than 20,000 droplets coalesce into just a few hundred within four to six hours. Unexpectedly, the total area of the detected droplets is not constant. Instead, there is a growing oil slick at the edges of the system that does not participate in any coalescences. After a subsequent look at the dynamics of single droplets, we suggest an explanation for the shrinking droplet area and support it with studies on the droplet area before and after an individual coalescence in section 4.2, showing that there is no oil mass movement into the third dimension (height). In section 4.3 we study aggregate measurements from our experiment. Surprisingly, despite being prepared identically, different time series vary strongly in their quantitative measurements, and sometimes even in their qualitative behaviour. Common, however, is an intriguing feature: the droplet size distributions become bimodal: While a first mode of small droplets accounts for a majority of the number of droplets, a second mode contributes almost all of the droplet area. In section 4.4, we use this observation to establish that the experiment dynamics go through three temporal regimes, and propose a mathematical description for the time evolution of the droplet size distribution.

4.1 PHENOMENOLOGY

Before we dive into detailed evaluations of our data, let us take a look at what the experiment can tell us at first sight. Below, we consider raw and processed images from a typical time series (in fact, from our very first run with the final parameter set). We find the coalescence behaviour familiar from the Mensa salad bowl that started it all: beginning with a large number of small droplets, we arrive at a small number of large droplets after a few hours. However, a number of very small droplets remain, contributing the bulk of the total number of detections, but almost nothing to the total detected area. Counter-intuitively, the total area of oil droplets constantly shrinks. The oil gathers in a constantly growing oil slick which touches all

system borders, thus representing a surrounding to all other droplets. The slick does not participate in the percolation process.

4.1.1 *Coalescence*

A time line from a typical experiment run and the corresponding detections from our image processing are shown in figs. 19 and 20. In the very first image, we detect about 33,000 oil droplets. A single one is larger than 10 cm², and 99 % are smaller than 1 cm². Starting from there, we see an initially fast, but increasingly slower coalescence process. After roughly 45 minutes, we find about 4,500 droplets that are all still smaller than 30 cm².

Three hours into the experiment, we see a little over 300 droplets, of which 17 are larger than 30 cm², and five (of these) are larger than 100 cm². 280 (i. e. 85 %) of the detected droplets are still smaller than 1 cm². Almost all of them are adjacent to large areas. We take a closer look at size distributions in section 4.3.

Figure 21 shows two close-up time lines of merging droplets. After a coalescence, and with enough time, the resulting droplet relaxes into a disk shape. Larger droplets exhibit a longer relaxation phase. Moreoever, coalescence events with droplets of similar size require longer relaxation times than those where a large droplet absorbs a smaller one. After all, the large droplet is already close to the final disk shape in the latter case.

4.1.2 *Inhomogeneities and the growing background*

Our experimental coalescence system is not perfectly homogeneous. First, we can spot different neighbourhoods within the system. These neighbourhoods vary in typical droplet size. In figs. 19 and 20, the contrast is particularly noticeable in the center of the system, where droplets are typically smaller than towards the outside.

Second, in some sectors near the edge of the system, the oil does not form small droplets which then enter the coalescence process. Rather, not particularly well-defined oil slicks are formed (cf. fig. 22). These slicks quickly connect and form a large slick that touches the edges of the system. With time, the slick retracts towards the edges, and the oil droplets are pushed into the center.

We declare this bordering oil slick as *background*, and its complement (i. e. the part of the system that contains "regular" oil droplets) as *active area*. From the images, we make the following observations by eye:

(i) The background does not directly interact with droplets in the active area.
(ii) The background experiences strong growth with time.

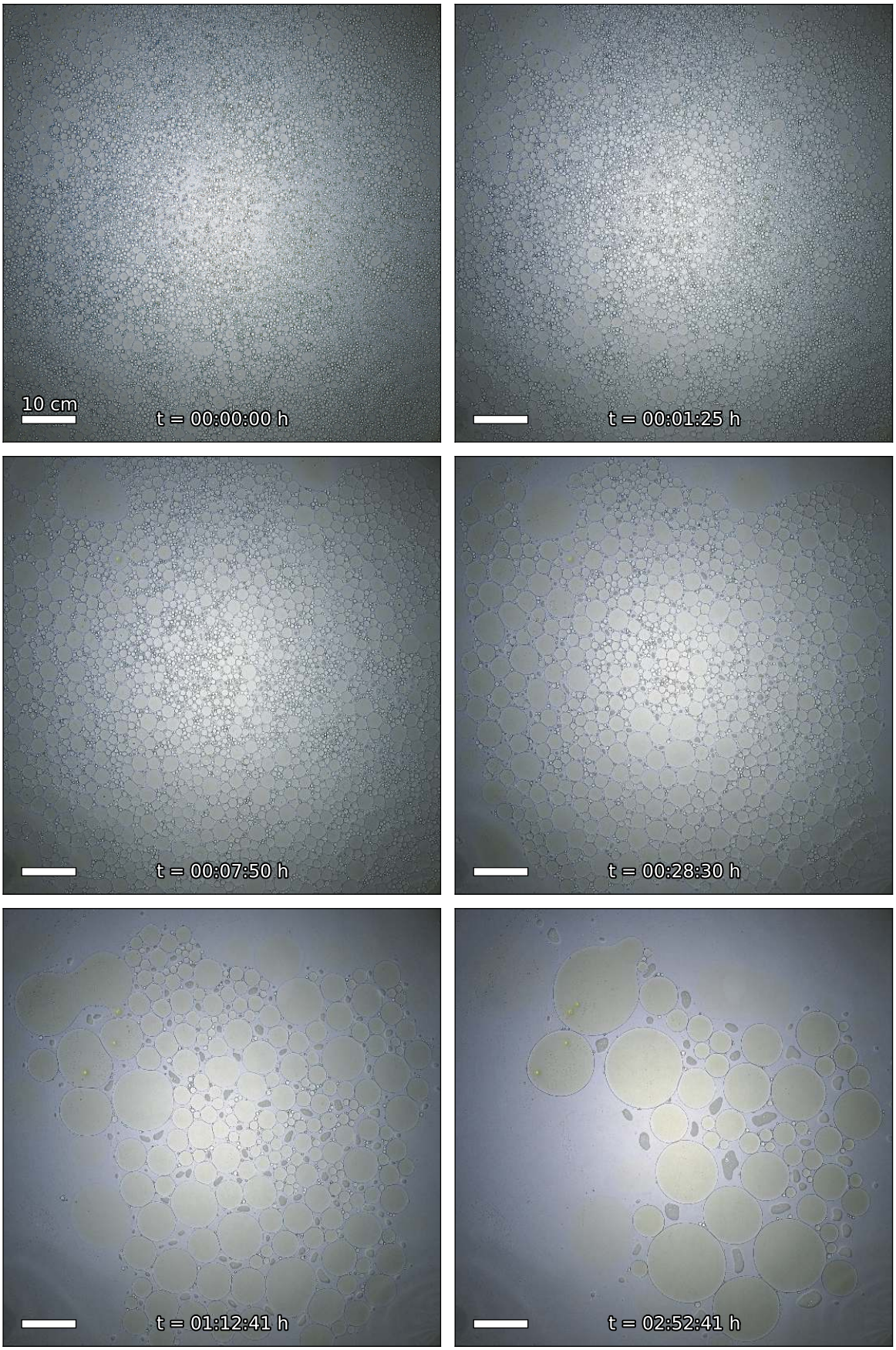


Figure 19: Raw images from a typical time series. Starting from roughly 33,000 (detected) oil droplets, we find initially fast, then increasingly slower coalescence, with a corresponding increase in typical size of the largest droplets. With advancing time, droplets gather in the area towards the middle of the setup, and darker areas separating the droplets become more and more prominent: aggregates of stable air bubbles.

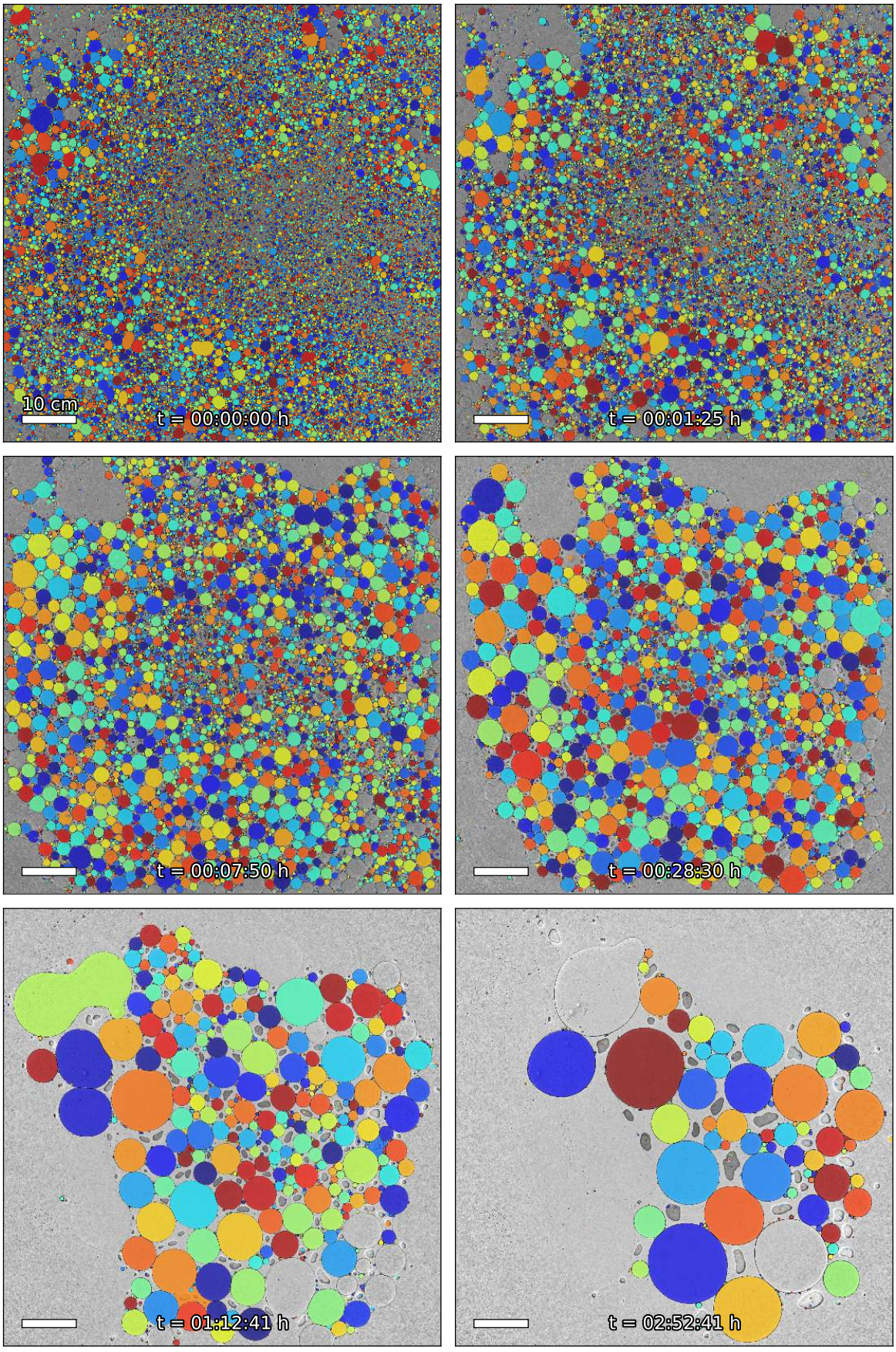


Figure 20: Areas detected as droplets from a typical time series (shown in fig. 19). Colour is solely for visual distinction of neighbouring droplets. It has no quantitative meaning. A few oil droplets missed by our image processing can be discerned. They arise especially near the corners of the setup, where illumination was limited (cf. fig. 19). Every detection touching the border of the image is declared background and discarded (and not coloured above).

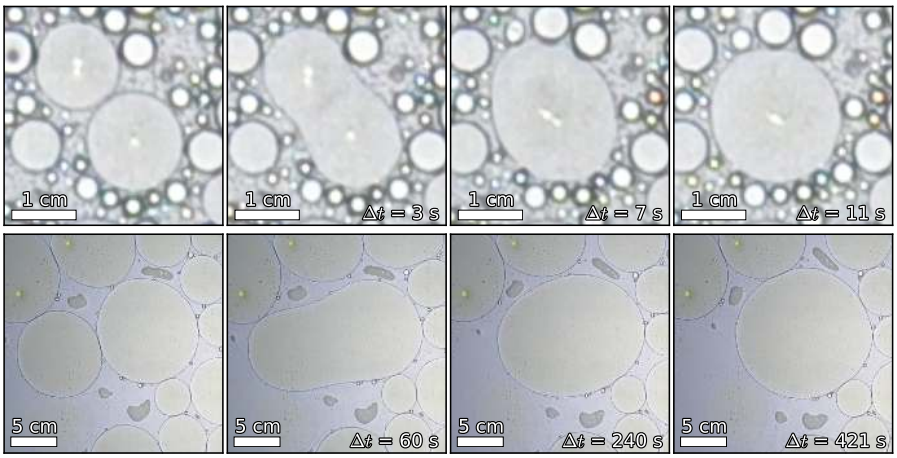


Figure 21: Time lines of typical coalescences. After a merging event (second column), the resulting droplet relaxes into a disk shape (final column). Larger droplets take a longer time to relax: *top*: droplets with initial diameters of about 1 cm take 11 seconds to relax; *bottom*: droplets with initial diameters around 5 cm need roughly 400 seconds to relax.

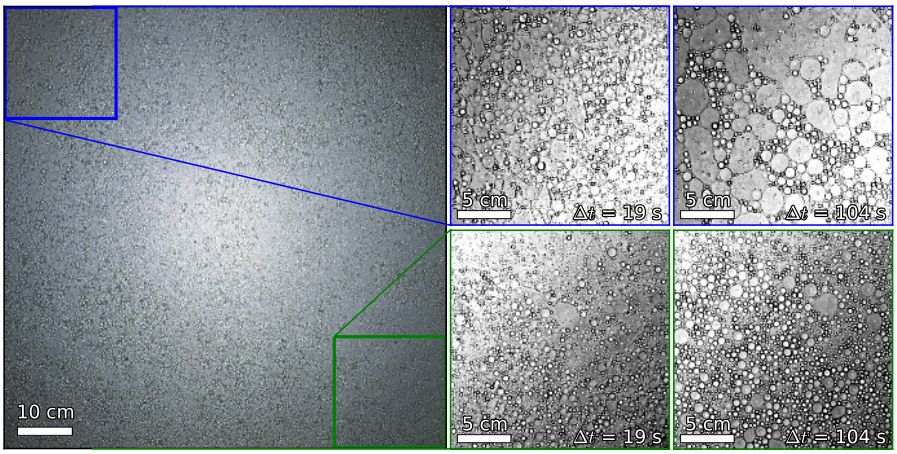


Figure 22: Background formation. After the initial stirring (see section 2.3) most parts of the system relax into small oil droplets (green). Some parts, however, form oil slicks that quickly connect, move towards the edges, and do not participate in the coalescence process (blue). We declare this large, surrounding oil slick as *background*.

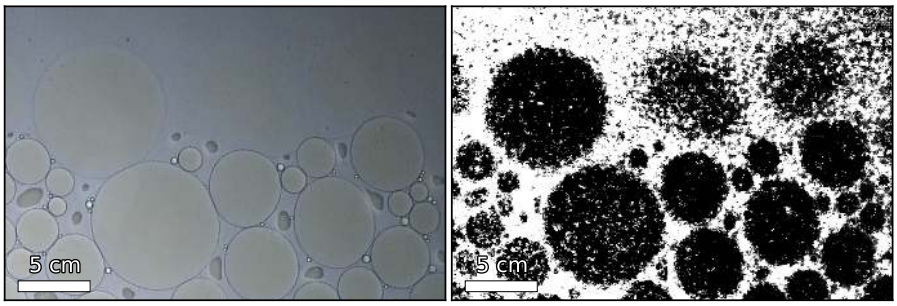


Figure 23: Droplets in active area and background. The image on the right shows the same crop from a raw experiment image as the one on the left. Contrast was strongly enhanced by selecting only the yellow-lime hues of the oil droplets in hue-saturation-value (HSV) colour space, then encoding the saturation channel as grey-scale. While it is possible to find droplets in the background (cf. top three clusters in right image), they have much lower contrast than droplets in the active area (cf. droplet edges in left image).

(iii) There are droplets in the background. These have a much lower contrast than droplets in the active area, and our image processing is incapable of reliably detecting them (cf. fig. 23).
(iv) The background droplets sometimes interact with each other, and very seldom they even interact with the active area.

Due to the unreliability of their detection, we ignore all droplets in the background, and consider their interactions with the active area as experimental noise.

In fig. 24, we see total droplet area for five experiment runs. Note that this area is not the same as the active area, which also includes the empty spaces between inner droplets (this area is not part of the background). Over the course of one experiment run, the total droplet area quickly reaches a peak of roughly 4,000 cm² (i. e. 40 % of the system area), about ten or twenty minutes into the experiment. Subsequently, it decays slowly to about a third or fourth of this peak value. Naturally, the total oil *volume* is a conserved quantity in our system.

Both the initial increase and slow decay in active droplet area have no intuitive explanation. For the increase, we see three possible causes:

(i) The detection limits and errors for small droplets, and the large relative error in area at sizes slightly above the limit, possibly systematically biased towards smaller areas.
(ii) The fact that the oil was freshly stirred (see section 2.3) and is still in the process of forming small droplets.
(iii) There might be an unseen oil film that is not part of the droplets, and that facilitates transport of oil from the droplets to the background surrounding the active area. This reduces tension on the oil droplets in the center and allows them to expand.

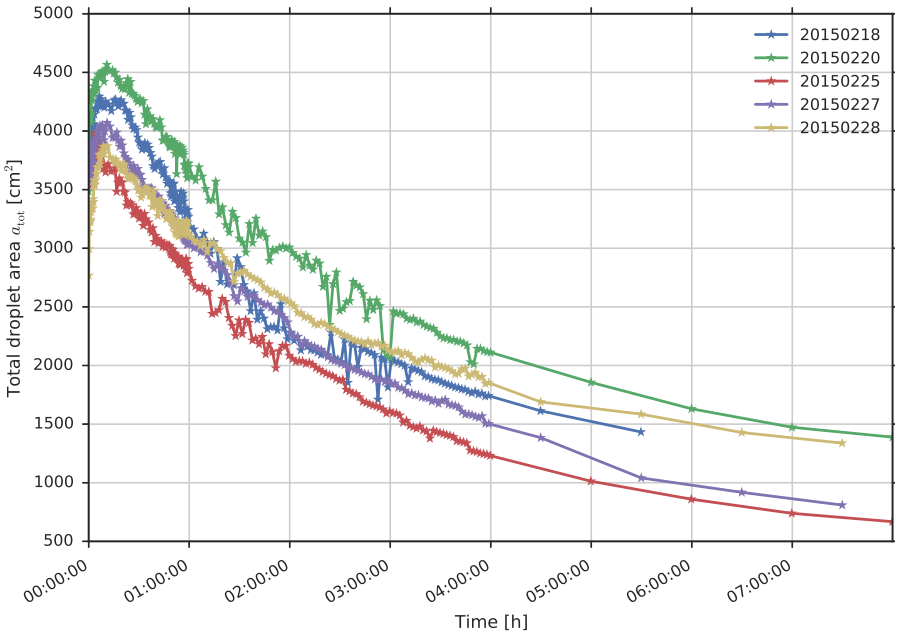


Figure 24: Time evolution of the total detected area. After an initial rise for about ten to twenty minutes, the area of all detected oil droplets decreases continuously to about a third or fourth of its peak value. At the peak, about 4,000 cm² of oil droplets are detected, i. e. about 40 % of the total system area is covered by droplets.

The subsequent decay could be the result of one or both of the following causes:

(i) With larger droplets, some oil moves into the third dimension (height), invisible to our optics.
(ii) Except for coalescence, all droplets experience constant shrinking. The oil moves into the background by some mechanism invisible to the cameras.

The aim of the subsequent detailed analysis in section 4.2 is to sort out which of these mechanisms are active.

4.1.3 *Surface pollution*

Besides the growth of oil droplets, there are also several growing dark areas visible, particularly in the last two images of fig. 19. These are aggregates of stable air bubbles that arise from our preparation of the initial conditions. Albeit the identical preparation of the experiments, their number and size varies between different runs of the experiment. We can make two observations based on our images:

(i) In the course of time, the dark aggregates gather towards the center of the setup, much like the oil droplets.

(ii) The bubble aggregates have a repulsive interaction with the oil droplets.

Given these observations, it seems reasonable to assume that they influence the dynamics of a single experiment in a fashion over which we have no control.

4.1.4 *Detection quality*

Due to the vast number of droplets, especially at early times, it is hard to quantify the quality of our droplet detection. Visual inspection of figs. 19 and 20 gives the impression that almost all medium- to large-sized droplets are correctly detected. There are only few false negatives. They arise near the edges of the system due to the far from optimal illumination. Assuming that the droplet neighbourhoods (see section 4.1.2) are distributed uniformly throughout the system, this should not affect our aggregate statistics (section 4.3) in the long run. At very late times, few false positives in the free space between droplets occur.

Figure 25 allows to judge the performance of the detection for small droplets. As the FCD algorithm works on gradients, it has difficulties in detecting droplets whose edges have low contrast to the background (and the inner part of the circle). Overall, FCD performance is good nonetheless, with no perceived bias towards missing droplets of particular sizes. The number of false positives within the air bubble aggregates (see above) is negligible.

4.2 DROPLET DYNAMICS

In the previous section, we saw that the total detected droplet area is not a constant: it rises by about a third of its initial value within 15 to 20 minutes, then slowly decays to about a third or fourth of its peak for the remaining six to eight hours of the experiment (cf. fig. 24). In this section, we investigate the mechanics underlying these observations. We will see that – contrary to droplets in the experiment – the first oil droplet deposited onto a fresh setup quickly expands to cover the water surface. A second droplet then lives on top this first layer (instead of blending into it), and expands much slower.

These observations from single droplets suggest that, in the experiment, a thin oil layer covers the complete water surface. This layer provides a mechanism that continuously transports oil from droplets into the background. We critically inspect this idea by examining the behaviour of droplets in the experiment in the absence of coalescences, and by examining the difference in area of a droplet resulting from a coalescence and the total area covered by the two droplets prior to the coalescence. Applying corrections implied by the oil-layer theory yields a master plot with a modified area measurement

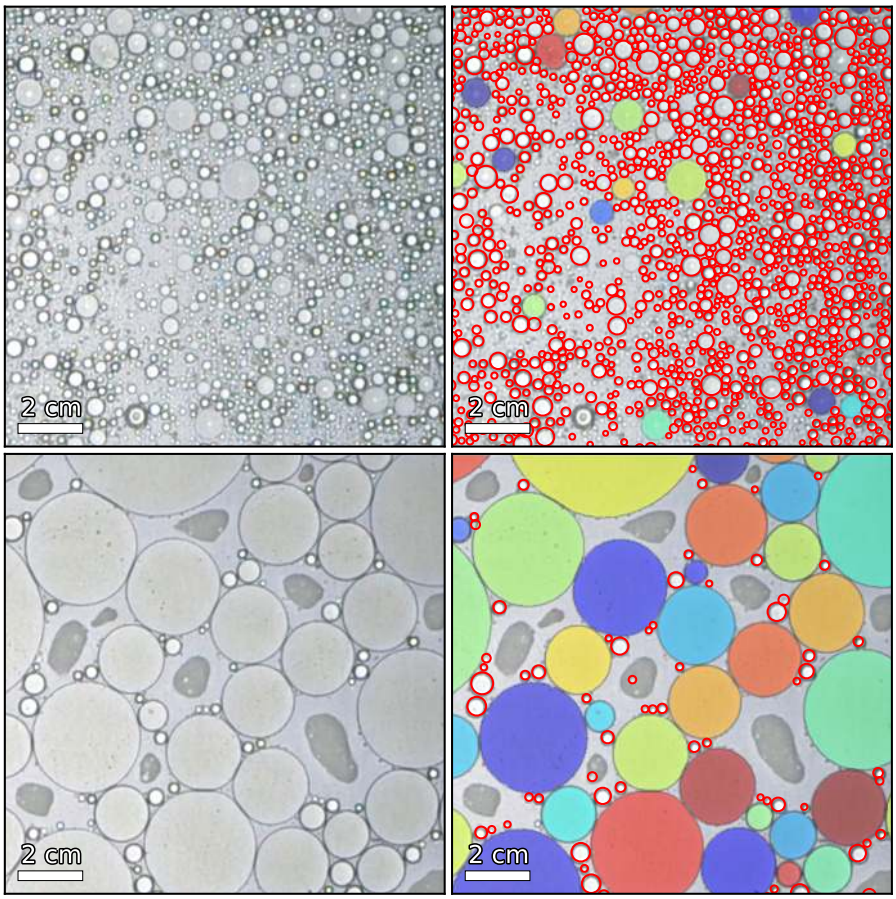


Figure 25: FCD performance. Red circles show detections from the FCD algorithm, while coloured areas are from the threshold detection methods geared towards large areas. At early times (top), the circle detection algorithm has problems with circles whose edges have low contrast (since it works on gradients), but performs reasonably good overall. Except for the smallest droplets, where we run into problems with the image resolution, there seems to be no bias towards missing droplets of certain sizes. In later images (bottom), there are few false positives within the dark areas (air bubble aggregates), and very few missed droplets.

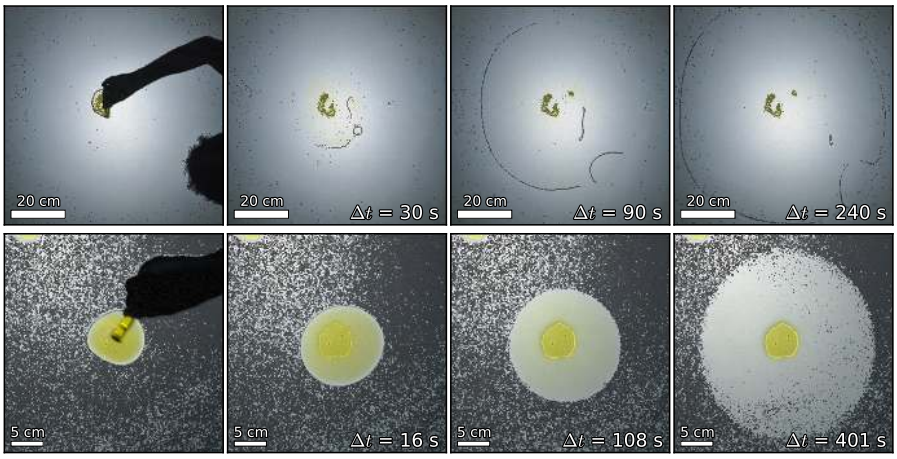


Figure 26: Expansion of single oil droplets. A small amount (20 ml) of oil is deposited onto a setup prepared identically to our experiments (top row). The droplet quickly expands to a diameter of 20 cm after half a minute (second column), and covers almost the complete setup after four minutes (final column, $\varnothing \approx 90$ cm). A second droplet deposited onto the same spot expands much slower (second row), from a diameter of $7 - 8$ cm after sixteen seconds (second column) to a diameter of $20 - 25$ cm after six and a half minutes (final column). The droplets' expansion is opposed to observations from the coalescence experiments, where droplets in the active area constantly shrink (excluding coalescence events). Due to the thinness of the droplets, contrast was very strongly enhanced in both time series: For the top row, the positions of the strongest gradients, marking the droplet's edge, are overlaid in grey. In the bottom row, the non-yellow-lime colours of the background are greyed out. The strong yellow spots in the centers of both image series are oil aggregates that are pinned to the glass bottom of the setup after pouring them in due to the low water height.

where both the area of a droplet in the absence of coalescences and the total area during coalescence are conserved.

4.2.1 *Single oil droplets*

Figure 26 shows the behaviour of single droplets deposited onto a setup prepared identically to our experiment (water and surfactant). The first droplet deposited immediately begins a fast expansion: after 30 seconds its diameter has grown to about 20 cm, and after four minutes a thin oil layer covers almost the complete setup. A second droplet deposited on top of this layer expands much slower, growing to a diameter of about 25 cm after six and a half minutes.

The observations from the single droplet experiment oppose observations from the coalescence experiments. Figure 27 shows the time evolution of the size of the largest droplet. While we can clearly see

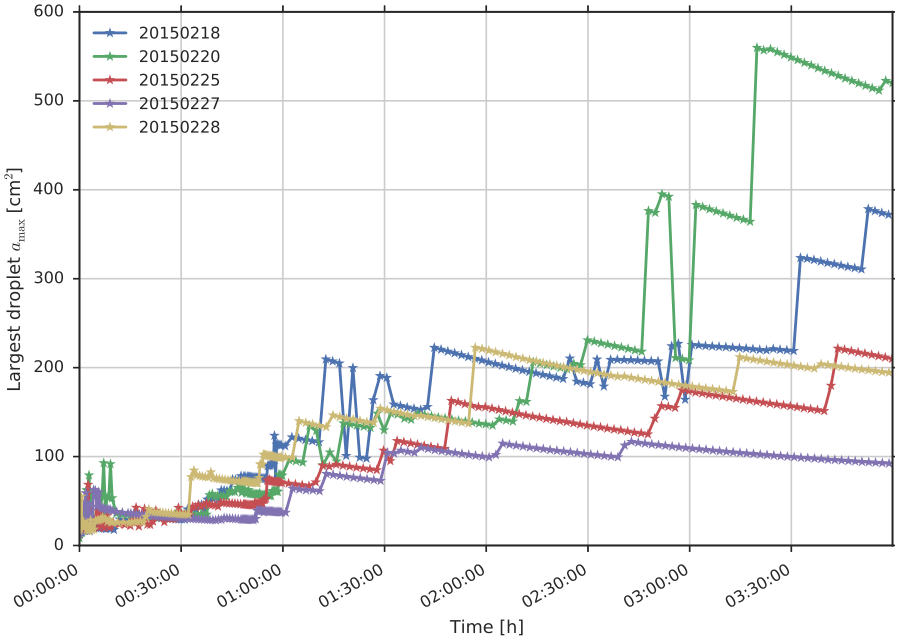


Figure 27: Evolution of the largest droplet. Aside from the expected jumps that arise from coalescence events, there is another striking phenomenon: the largest droplets shrink in between the droplet merge events. This shrinking is not related to the droplet relaxing into a disk shape (cf. fig. 21), which happens on much shorter time scales. It opposes observations from our single droplet experiments, where the droplets continuously expanded. The rate of area decrease seems to grow with the area (particularly well visible in the green (20150220) and red (20150225) data series). Large back-and-forth jumps are artifacts resulting from detection errors.

the expected big jumps resulting from coalescence events, another feature in these time series catches the eye: In between the coalescences, the largest droplets do not expand. Rather, they shrink. This shrinking is different from the droplet relaxing into a disk shape after a coalescence (cf. fig. 21), as the relaxation times are much shorter.

In section 4.1.2, I suggested that the decay of the total detected area may be traced back to two independent mechanisms: oil moving into the third dimension (i. e. a size-dependent droplet height profile, contrary to a pancake shape), and oil being invisibly transported from droplets into the background. While both mechanisms will be dealt with in section 4.2.2, we can make a first, rough check for the latter mechanism by normalising the observed size of the largest droplet with the (changing) total droplet area.

Indeed, *in the absence of coalescences*, the normalised droplet sizes (fig. 28) do no longer show any significant droplet expansion or shrink-

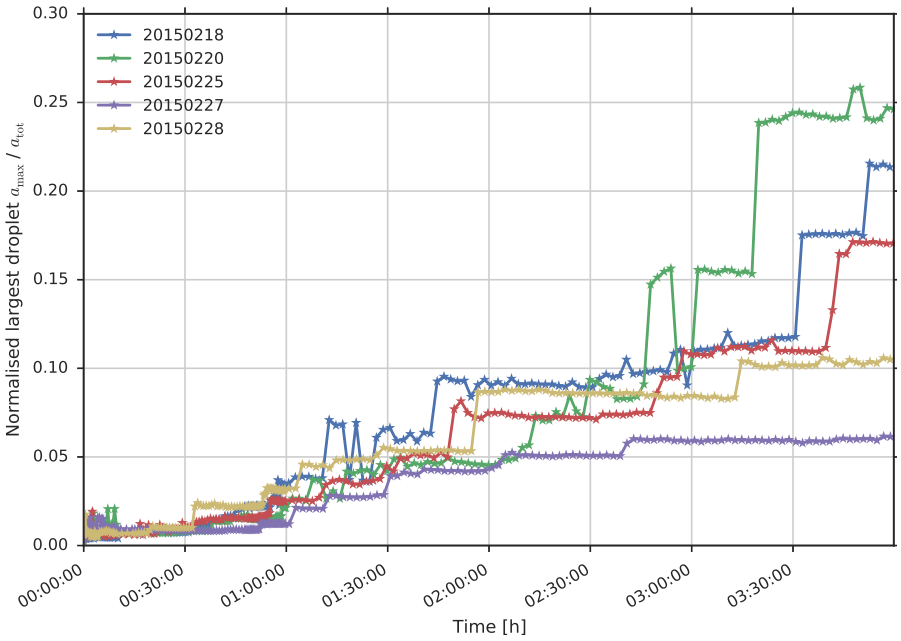


Figure 28: Evolution of the largest droplet, normalised by the total detected area. The droplet shrinking in between coalescences (cf. fig. 27) has vanished, such that a_i/a_{tot} is a constant, implying $\dot{a}_i \sim a_i$.

ing. Instead, the ratio $a_i(t)/a_{tot}(t)$ for any droplet i with area a_i seems to be a constant in between collisions. Considering

$$\begin{aligned} 0 &= \frac{d}{dt}\frac{a_i}{a_{tot}} \\ &= \frac{\dot{a}_i \cdot a_{tot} - a_i \cdot \dot{a}_{tot}}{a_{tot}^2} \\ \Rightarrow \quad 0 &= \dot{a}_i \cdot a_{tot} - a_i \cdot \dot{a}_{tot} \\ &= \dot{a}_i \cdot \sum_j a_j - a_i \cdot \sum_j \dot{a}_j, \end{aligned}$$

this implies

$$\frac{da_i}{dt}(t) \sim a_i(t) \cdot f(t). \tag{1}$$

Again, it should be stressed that this describes the droplet area change behaviour only in between coalescences.

The above considerations yield

$$\begin{aligned} &\frac{d}{dt}a_{tot} \sim a_{tot} \cdot f(t) \\ \Rightarrow \quad &a_{tot} = a_{tot}(t_0) \cdot \exp\left(\int_{t_0}^{t} d\tau\, f(\tau)\right). \end{aligned} \tag{2}$$

For a first approximation, we assume that $f(t)$ is a constant, such that

$$a_{tot}(t) = a_{tot}(t_0) \cdot e^{\alpha t} \tag{3}$$

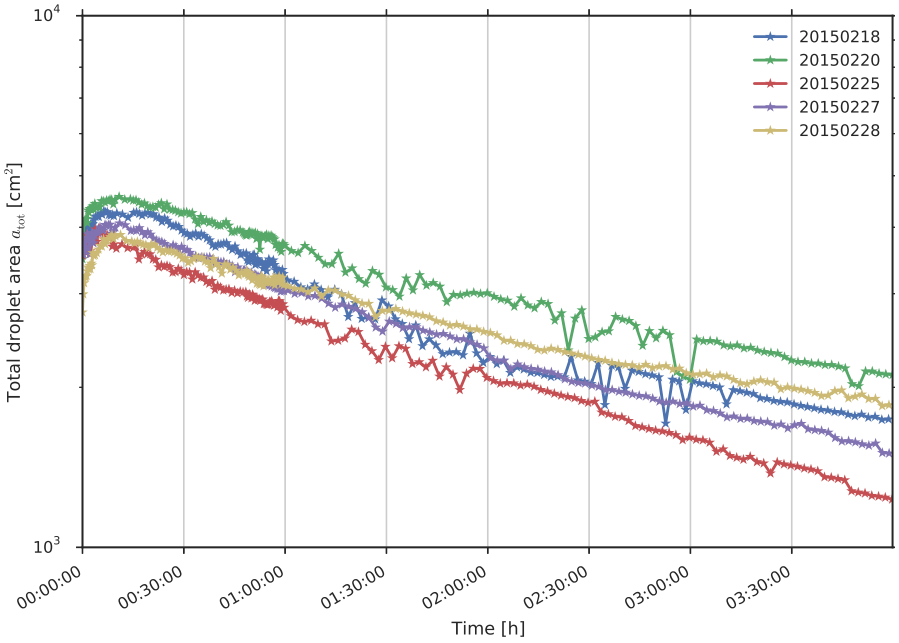


Figure 29: Evolution of the total detected area (on a logarithmic scale). The total area seems to roughly follow an exponential decay, as expected from fig. 28 and eqs. (1) and (2), but does not perfectly fall onto a straight line on the logarithmic scale. This mismatch might arise from both inaccuracies in our approximation or from effects associated with the droplet merging.

That this approximation is reasonably good, but not completely accurate, can be seen in fig. 29. The deviations of the total detected area from a perfectly exponential function might arise from both errors in the assumption of a constant $f(t)$ as well as from effects associated with the droplet merging.

With these insights, we propose the following explanation for the total detected area decay: As soon as there is a background area at the edges of the system, there is a physical force driving oil into this background. This force can mainly be traced back to the line tension of the oil, as

(i) the oil droplets in the active area can decrease their perimeter by shrinking, and
(ii) the background can decrease the perimeter of its "inner hole" (i. e. the active area) by growing.

The oil transport mechanism is provided by a thin oil layer covering the complete active area (as we have seen in the single droplet experiment). This agrees with the oil droplet mass loss being roughly proportional to the droplet area (eq. (1)), as this area is the droplet's interaction interface with the layer.

The idea of an oil layer that invisibly transports oil from the active area into the background could furthermore explain the initial rise

in total detected area (cf. section 4.1.2 and fig. 24). We sugeest that in the early stage of the experiment, when the background is still in the process of forming, almost all of the oil is in the active area. This introduces a strong tension on the oil droplet edges, moving oil into the third dimension, therefore reducing the droplet area visible to the camera. As the background forms and collects oil from the active area, this line tension declines, allowing the droplets to expand. Later, this effect is superimposed by the (stronger) oil droplet mass loss through the (now established) transport mechanism.

4.2.2 *Coalescence radii*

In the previous section, we have proposed that the decay in total detected area is caused (at least partially) by oil being transported into the background through a thin oil layer covering the complete setup. In this section, we investigate the second mechanism proposed in section 4.1.2: the droplets might have a height profile that is dependent on their size (opposed to a pancake shape). In this case, with growing droplet size, oil would move into the third dimension (droplet height), thereby hiding from the camera and our image processing. This mechanism can easily be tested: if the droplets are not pancake-shaped, i. e. if oil moves into the height profile of the droplet, the area of a merged droplet would differ from the total area of its two ancestor droplets.

In section 4.1.1 and fig. 21, we saw that after a coalescence event, the droplet relaxes into a circular shape. In the following, we inspect the *radii*, not the *areas*, of droplets involved in coalescence events. This is motivated by the fact that the height profile which we are interested in is only consistent between different droplets of the same size when they are in a relaxed state. In other words, *of course* the total area of a merged droplet equals the sum of its two ancestral droplets *immediately after their edges connect*. We are, however, interested in the total area after the droplet has undergone this immediate, fast area change, i. e. when the droplet has relaxed into a round shape. In this relaxed case, a droplet's area and radius are exchangeable ($a = \pi r^2$), and we stick to using radii to constantly remind us that we are looking at an equilibrium state.

Figure 30 shows the ratio of merged area (r_3^2) and total initial area ($r_1^2 + r_2^2$) after a coalescence event. Except for the ratios for very small droplets, where relative errors are very large, there is a distinct trend in the data: the area ratio decreases with growing droplet size. Coalescences of small droplets lead to a total area increase of up to roughly 5 %, whereas coalescences of large droplets lead to a total area decrease of up to 3 %. This observation might be a hint towards a size-dependent droplet height profile. However, in section 4.2.1, we saw that droplets continuously lose mass, and that – independent

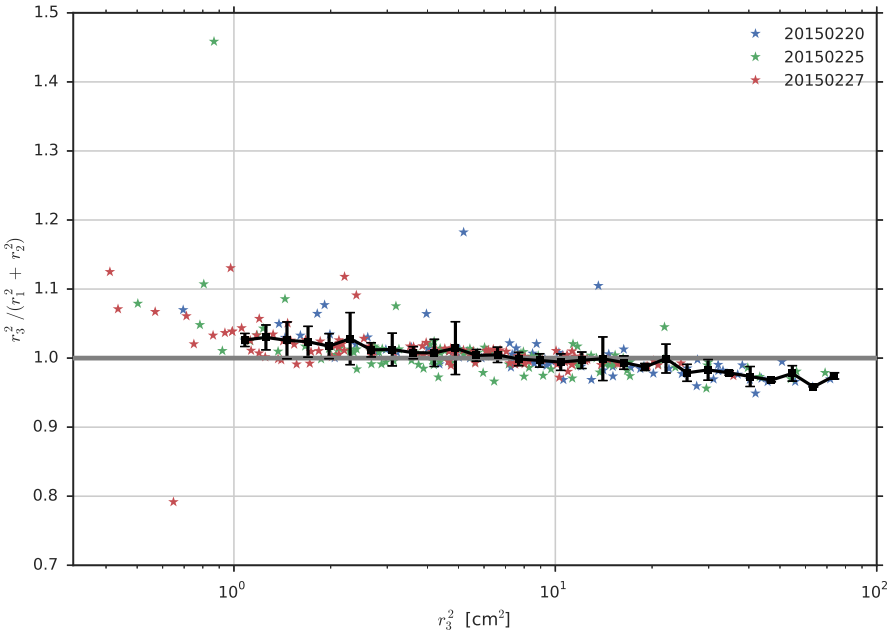


Figure 30: Ratio of total initial and merged area after a coalescence of two disk-shaped droplets with radii r_1 and r_2 into a disk-shaped droplet with radius r_3. Stars represent data from individual merging events. The black line shows the average and standard deviation of events with similar final radius r_3. There is a small, but distinct trend in the ratio: coalescences of small droplets lead to a total area increase of up to roughly 5 %, whereas coalescences of large droplets lead to a total area decrease of up to 3 %. Ratios for very small droplets vary wildly due to the large relative error in size measurement and the large relative impact when additional droplets took part in the coalescence and were missed in the visual inspection.

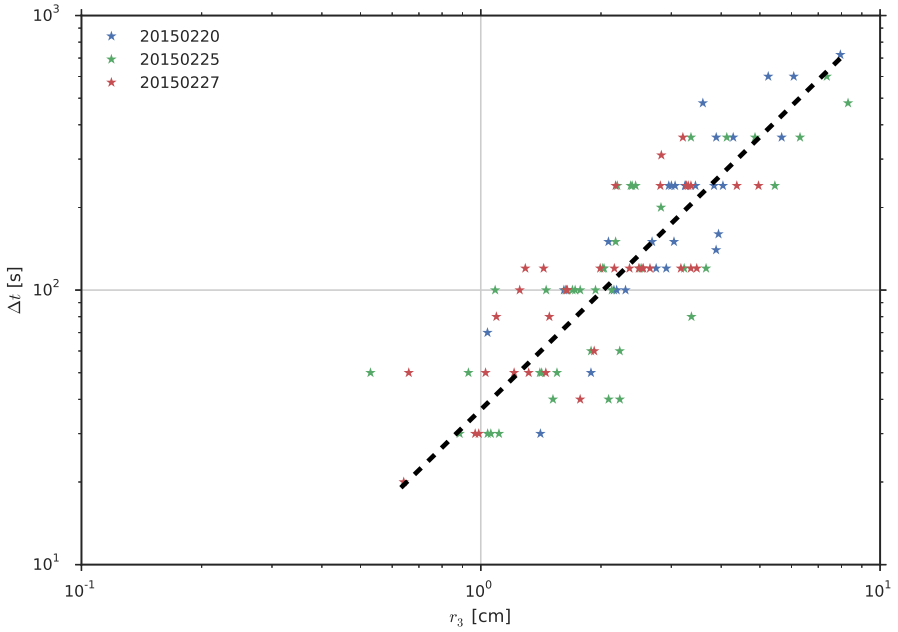


Figure 31: Relaxation time for droplets of different sizes. When two droplets coalesce, the resulting droplet relaxes into a disk shape (cf. fig. 21). The relaxation time clearly increases with the size of the resulting droplet, and seems to do so in a roughly power-law fashion. Relaxation times range from 1.5 seconds for droplets with a radius of 0.5 cm to 10 minutes for droplets with a radius of 8 cm. The power law can be fitted reasonably well with $\Delta t \sim r^{4/3}$ (dashed line). However, the small range of our measured radii (about one decade), our time resolution limits, and the large variance within the data forbid drawing quantitative conclusions.

of our oil layer transport theory – this should be compensated for by normalising areas with the total detected area. As the radius r_3 of the resulting droplet is measured at a later point in time than the radii r_1 and r_2 of the two ancestral droplets, in fig. 30, we have compared radii that are in fact *not comparable* in the context of height profile investigation.

While the time difference, and the corresponding change in total detected area, between measuring the radius of the resulting droplet and the radii of the ancestral droplets might be negligible for small droplets, its impact grows with increasing droplet size. Figure 31 shows the relaxation time, i. e. the time difference between measuring r_3 and $r_{1,2}$, plotted against the radius r_3 of the resulting droplet. The time that a droplet needs to relax strongly increases with the droplet's size, from less than half a minute for droplets with a radius of 1 cm to ten to twenty minutes for droplets with a radius of about 8 cm. While a power law can roughly be fitted to the scatter data by eye, no quantitative conclusions can be made due to the small range of measured radii (one decade), the time resolution limits, and the large variance within the data.

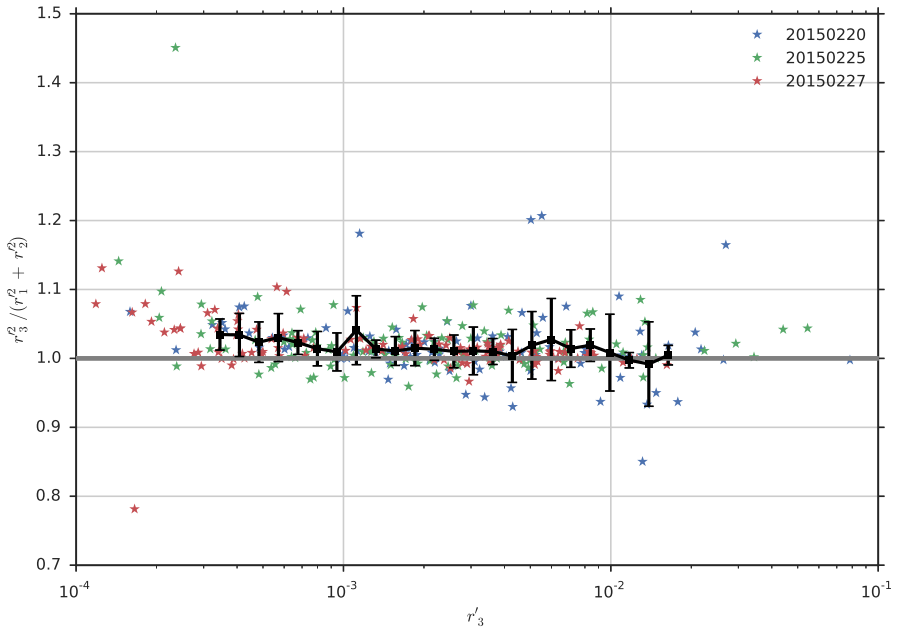


Figure 32: Ratio of total initial and merged area after a coalescence of two disk-shaped droplets with radii r_1 and r_2 into a disk-shaped droplet with radius r_3. The radii have been normalised with the total droplet area (cf. fig. 24), i. e. $r_i' = r_i/\sqrt{a_{tot}(t_i)}$. While the trend of coalescences of large droplets resulting in slightly smaller areas (cf. fig. 30) is not completely canceled by the normalisation, the total resulting (normalised) droplet area is no longer smaller than the sum of the two initial areas. The variance within the data increases slightly with the normalisation.

However, the increasing relaxation time can explain the trend seen in fig. 30: The larger the resulting droplet, the larger the oil mass loss resulting from the continuous shrinking. For analysing the droplet height profile change, we need to *suppress* this effect of the increasing relaxation time, i. e. individual shrinking of isolated droplets. In section 4.2.1, we saw that this can be achieved by normalising all droplet areas with the total detected area. Figure 32 shows the ratio of initial and merged area for radii that have been normalised with the total droplet area at the moment of measuring, i. e.

$$r_i' = r_i/\sqrt{a_{tot}(t_i)}, \tag{4}$$

where t_i is the time at which r_i has been measured. Indeed, the normalised data does not show a distinct trend: a horizontal line could easily be fitted through all error bars, potentially also because the normalisation slightly increases the scatter in the data.

The continuous oil movement towards the background explains the *trend* seen in fig. 30. However, while almost all error bars touch the unity ratio expected for a perfect conservation of area, almost all binned ratio averages (black dots in fig. 32) lie above 1.0. Again, this could be explained by assuming that a thin oil layer covers the

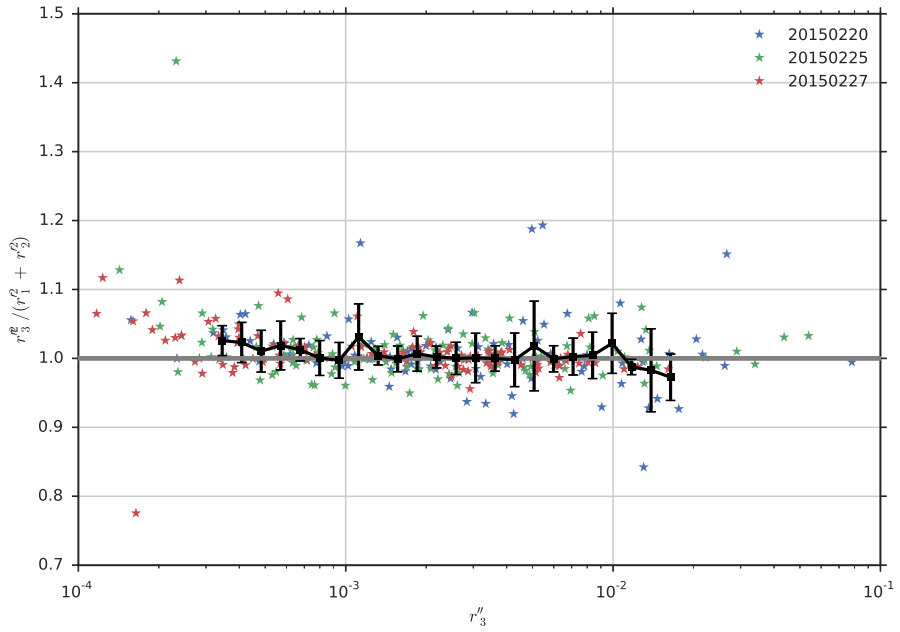


Figure 33: Ratio of total initial and merged area after a coalescence of two disk-shaped droplets with radii r_1 and r_2 into a disk-shaped droplet with radius r_3. The radii have been normalised with the total detected droplet area (cf. fig. 24), *and* with a fraction of the area that is overtravelled while the resulting droplet relaxes, eqs. (4) and (5). The free parameter α is set to $\alpha = 0.01$. With these corrections, the resulting area of an average coalescence varies less than 3 % of the initial area for almost all droplet sizes.

setup. During the relaxation process of a merged droplet, it moves over and onto an area that was not previously covered by droplets. The oil layer from this area is absorbed into the droplet, effectively increasing its oil mass and therefore its radius in the relaxed shape. We can test this hypothesis by introducing a second radius correction,

$$r''_3 = \left(1 - \alpha \cdot \sqrt{\frac{a_{\text{new}}}{a_3}}\right) \cdot r'_3, \tag{5}$$

where α is a free parameter related to the fraction of the oil layer that is actually absorbed into the droplet, and a_{new} is the area that was previously not covered by the two initial droplets. This correction maps the (normalised) radius of the resulting droplet onto the (normalised) radius that it *would have had*, had it not absorbed any oil from the oil layer.

Figure 33 shows the ratio of initial and merged area with both corrections, eqs. (4) and (5), applied. The free parameter α has been set to 0.01. Particularly in the range of median droplet radii, the binned ratio averages fall very close to unity, with typical deviations less than 1 %. However, it should be stressed that we had to introduce a free parameter to achieve this excellent agreement.

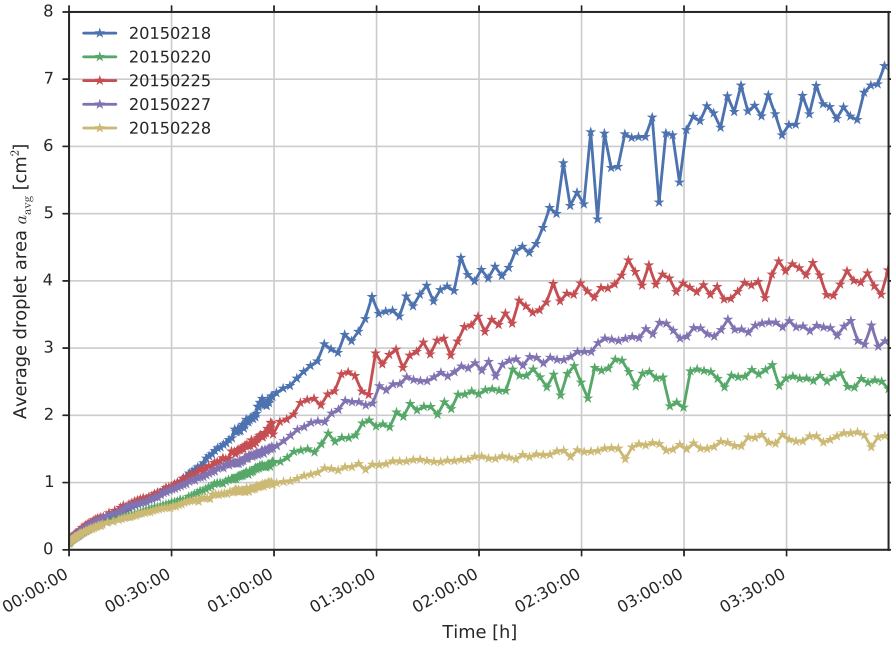


Figure 34: Average droplet size. In each run, the average droplet size grows initially, and it stabilises in the later stages of the experiment. While (expectedly) starting at the same average, different experiment realisations quickly diverge in average droplet area. The strong variability severely hampers calculation of ensemble averages.

The observations from this section further support the oil layer theory: Applying corrections that it implies, eqs. (4) and (5), allows the creation of a master plot where the area of a droplet resulting from a coalescence is equal to the sum of the two areas of the ancestral droplets. This rules out that the droplets have a size-dependent height profile. In all of the following analysis, we will therefore assume that the droplets are pancake-shaped. I. e. they are uniform in height except at the edges.

4.3 AGGREGATE MEASUREMENTS

4.3.1 *Droplet growth*

In fig. 34, we find the time evolution of the average droplet size: for all runs, this size initially undergoes continuous growth. Even though intuitive, this is not inevitable: both the total droplet area a_{tot} and the number of droplets N are constantly changing. If we had a large number of small droplets that do not take part in coalescences, we would expect the average droplet size to first grow, while the decrease in N through coalescence dominates, then shrink again while the decrease in total droplet area dominates. We will gain a better understanding of the average droplet area in section 4.4.

Figure 34 is exemplary for an observation we will make throughout many parts of our data evaluation: the identically prepared experiment realisations diverge not only quantitatively, but also qualitatively, and repeatedly even dramatically: the largest and smallest droplet size averages after four hours can differ as much as by a factor of four. The green data series (20150220) stabilises much earlier than other runs; on the other hand, the blue one (20150218) barely stabilises within the four hour time frame we look at. Similarly, in the previous sections, we saw that both the peak and the decay rate of the total detected area vary so much that data series cross in the combined plot (fig. 24), and that the largest droplet constitutes a quarter of the total detected area after four hours in the green data series (20150220), but only about 7 % in the purple one (20150227, fig. 28). This variability of the data prevents naïve averaging of experiment runs to find ensemble averages.

Figure 35 shows the droplet packing density, i. e. the ratio of total *droplet* area and total *active* area. In the first twenty minutes of all experiment runs, we see a rapid increase from under 50 % up to 70 – 80 %. This peak is held, or for some runs further increased, within the following forty minutes. One hour into the experiment, as droplets become increasingly bigger (cf. fig. 34), the packing density starts declining, and does so roughly linearly for the rest of the experiment. After four hours, all packing densities lie between 50 and 60 %. Just like with the total detected area (cf. fig. 24), the data series of different experimental realisations, in particular their decay rate, vary so much that their graphs cross.

4.3.2 *Coalescence speed*

In this section, we will study the evolution of the number of detected droplets, and the (related) evolution of its change, i. e. the coalescence speed $\frac{dN}{dt}(t)$. In fig. 36, we see the number of detected droplets as a function of time. All experiment runs start at a similar level (on a logarithmic scale) between 20,000 – 30,000 droplets and stabilise after about four hours. Unlike the start levels, the stabilisation levels differ significantly: from about 200 droplets (blue data series, 20150218) to about 1,000 droplets (yellow run, 20150228), a separation by a factor of five.

If we plot not only the number of detected droplets, but also the time on a logarithmic scale, as in fig. 37, we see that there is a crossover in qualitative behaviour about 1,500 seconds (50 minutes) into the experiment. After that crossover time point, all runs can roughly be approximated by a power law. The power-law-like behaviour lasts until the finite size effects take hold of the experiment, i. e. until the stabilisation of the droplet number. While this is the large majority of the experiment time, it is not even a decade on the logarithmic time

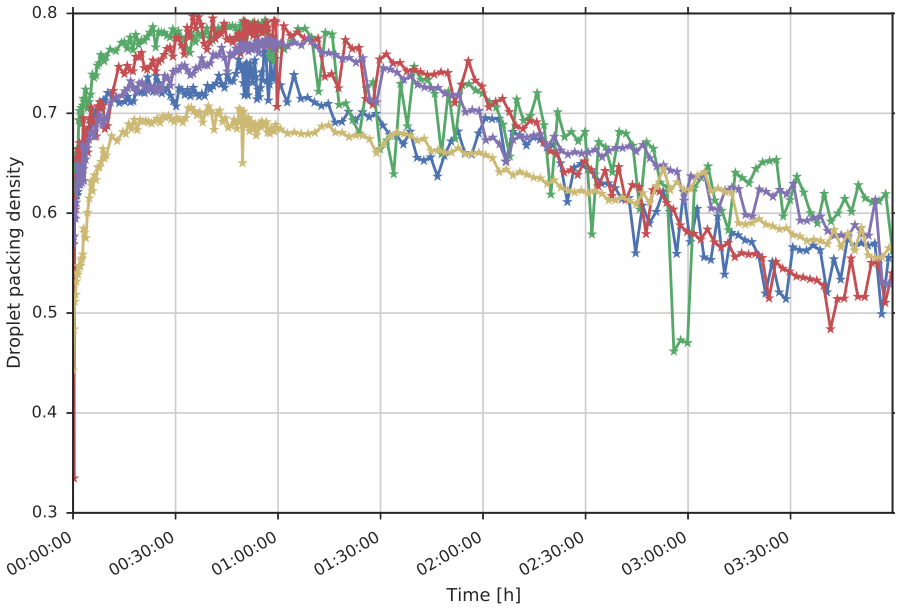


Figure 35: Droplet packing density. Starting at 40 – 50 %, after a very strong increase within the first twenty minutes, the fraction of the active area covered by droplets reaches a maximum at around 70 – 80 % after 45 minutes. For the remaining hours of the experiment, as droplets become increasingly larger, it declines in a roughly linear fashion. After four hours, the packing fractions of all experiment realisations lie between 50 and 60 %.

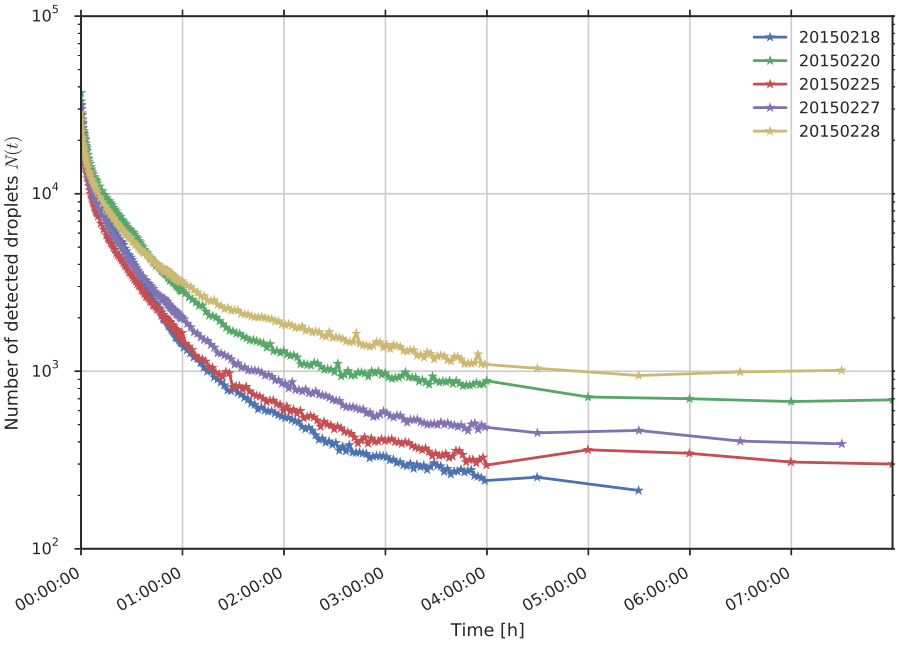


Figure 36: Total number of droplets. While all experiment realisations start at about the same number of droplets (on a logarithmic scale), they diverge with time. After four hours, the number of detected areas stabilises between 1,000 droplets and 200 droplets.

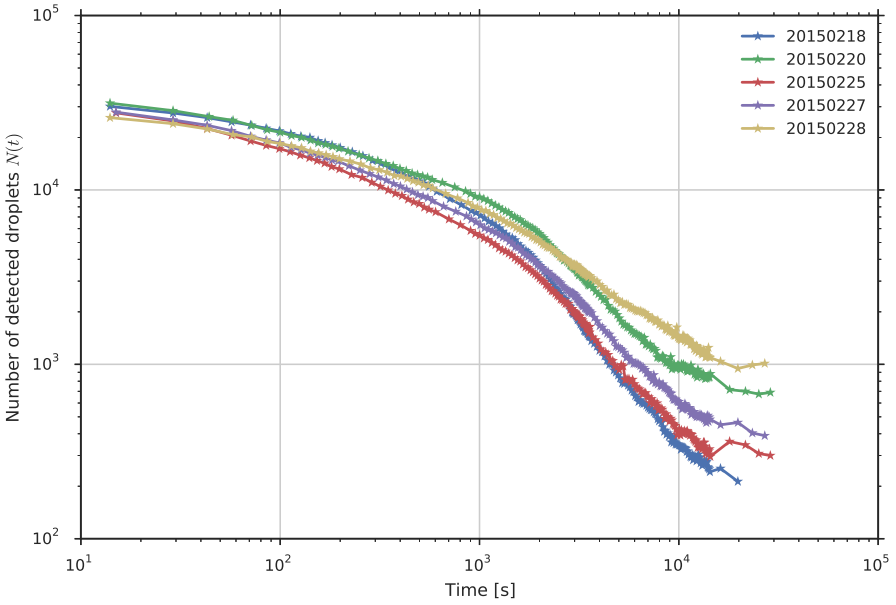


Figure 37: Total number of areas on double-logarithmic scale. After about 1,500 seconds, i. e. at maximum droplet packing density (cf. fig. 35), we see a crossover in the qualitative behaviour: from a logarithmic (cf. fig. 38) to power-law decay. However, this power-law decline lasts less than a decade before the total number of areas stabilises.

scale, and the power-law approximation should therefore be treated with great caution.

We can take a closer look at the early behaviour, i. e. before the crossover at 1,500 seconds, by plotting only time, but not droplet number on a logarithmic scale. In fig. 38, we have furthermore normed the droplet number by its initial value. The normed number of droplets, $N(t)/N_0$, show a convincing data collapse in the early stages of the experiment, where we see a rapid decline in the total number of droplets. After 1,000 seconds, the number of droplets has fallen to roughly 20 % of its initial value for all experiment runs. The data collapse can be approximated by a logarithmic function:

$$N(t)/N_0 = \ln\left(\frac{t}{t_0}\right)^{-a} \tag{6}$$

In fig. 39, we plot the (negative) coalescence speed, $-\frac{dN}{dt}$, against time. Again, we find a good collapse of all experiment realisations in the first one to two thousand seconds of the experiment. During this period (i. e. about two decades in the logarithmic time scale), the coalescence speed seems to decay in a power-law fashion. At later stages, where the total number of droplets is relatively small, the relative differences in number of droplets between data series are large (cf. fig. 37), and finite size effects become important, the coalescence speed data varies more wildly, both between different and within single experiment data series.

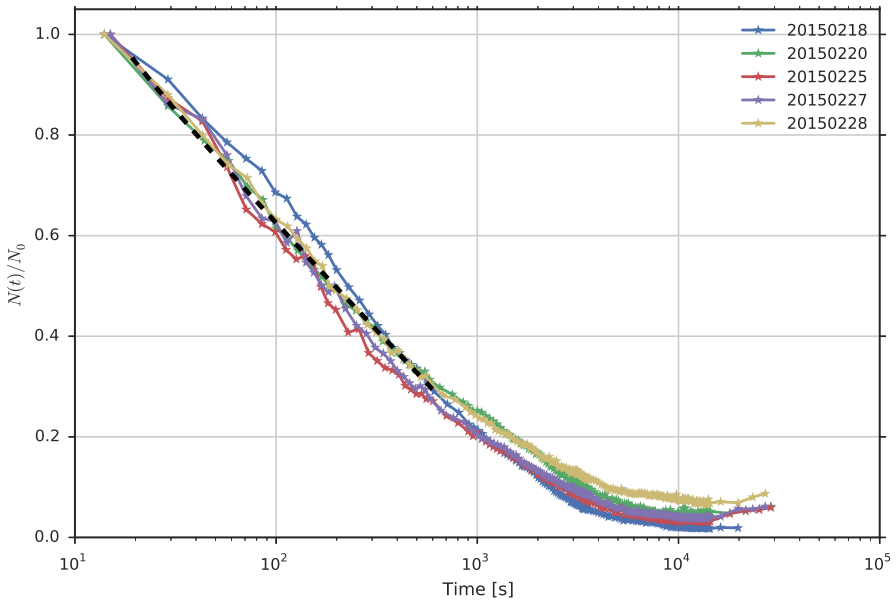


Figure 38: Normed total number of areas on logarithmic time scale. At the very start of the experiment we see a rapid decline in the number of detected droplets. All experiment realisations lose about 80 % of their initial droplets after 1000 seconds. In this very short time frame, the droplet number decay can be approximated reasonably well with a logarithmic function.

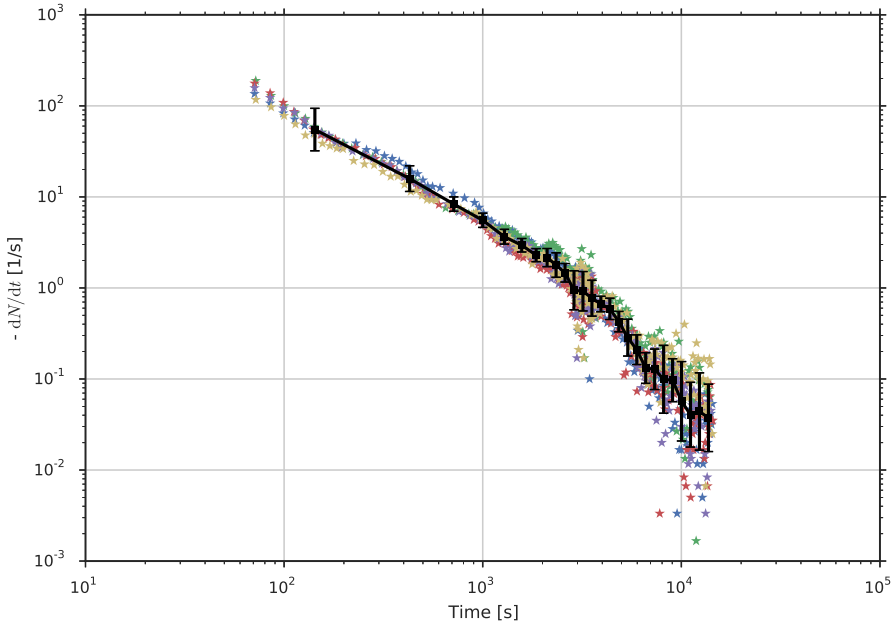


Figure 39: Coalescence speed as function of time. In the first 1,000 – 2,000 seconds, all experimental data collapses onto a power-law, while towards the end of the experiment, where finite size effects have an increasingly larger influence, the coalescence speed starts to vary more wildly.

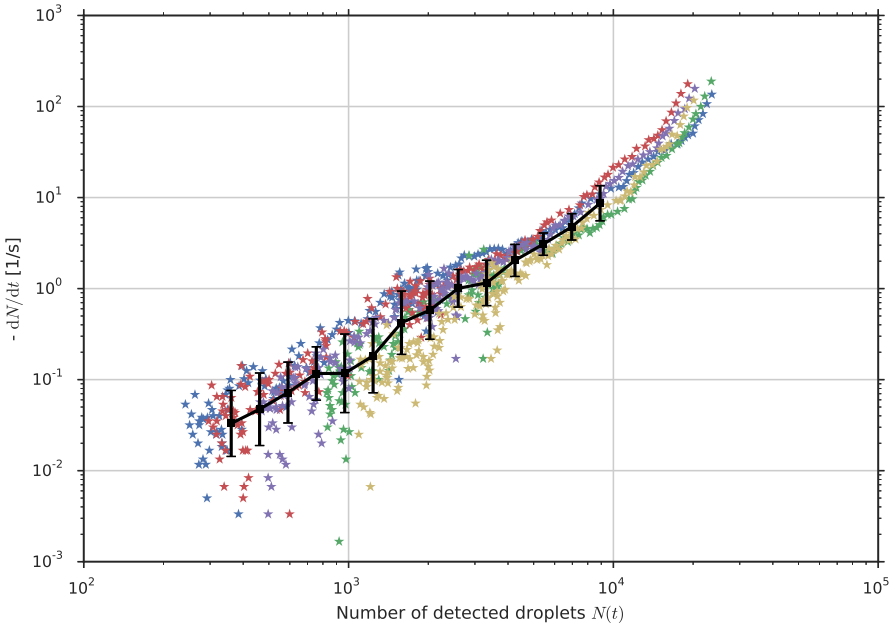


Figure 40: Coalescence speed as function of droplet count. While we find no data collapse as in fig. 39, we do get a better insight into the later stages of the experiment: when the droplet number decreases in a power-law fashion (cf. fig. 37), the coalescence rate also seems to vary with the droplet number in a power-law fashion.

To gain a better insight into the coalescence speed during the late stages of the experiment, we plot it against the number of droplets in fig. 40. Again, we find a large variation in the experimental data for late times, i. e. for small droplet numbers. This could possibly be explained through the very low coalescence rates at these times, down to a single coalescence per 100 seconds. Despite the relatively large variation, all data can be fitted quite well with a power law for small droplet numbers, ranging over about one and a half decades. This range of droplet numbers for which the power-law fit holds corresponds to the time span after the crossover in qualitative behaviour from fig. 37: At 1,500 seconds, we find around 8,000 droplets.

4.3.3 *Size distributions*

In section 4.1.2, we noted that the spatial distribution of droplet sizes is non-uniform: the experiment is characterised by local neighbourhoods comprised of similar-sized droplets. Here, we will take a look at the aggregate, i. e. non-local, distribution of droplet sizes: the evolution of the experiment's droplet composition on a global scale.

Figure 41 shows the probability density $p(a, t)$ of finding a droplet of size a for different points in time. In general, the evolution begins with a narrow distribution peaking at a very small droplet size of around 0.03 cm². This peak exists throughout all times, moving only

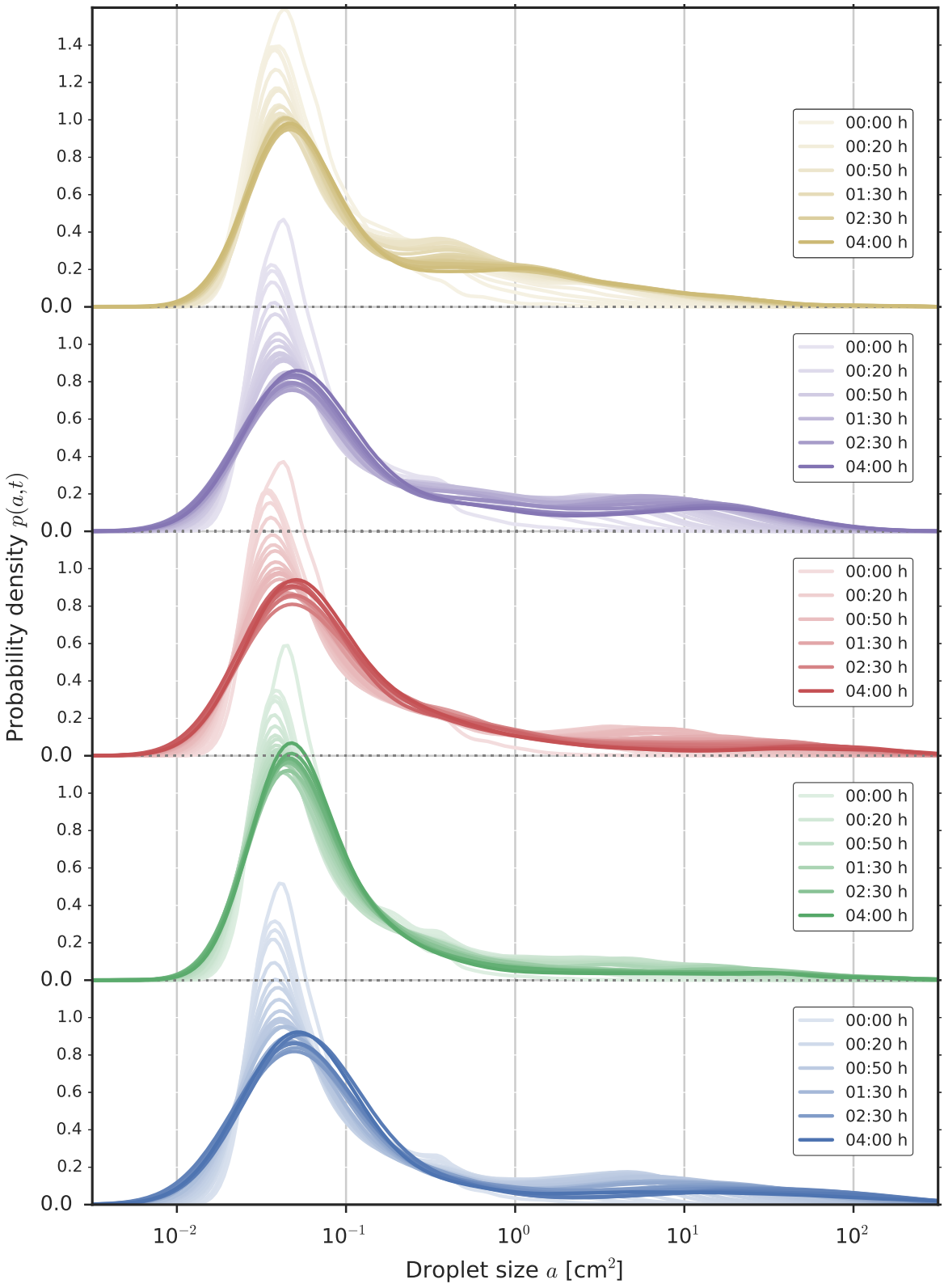


Figure 41: Probability density of droplet sizes. All experiment realisations start with a narrow distribution of small droplets, peaking at about $3 \cdot 10^{-2}$ cm². The peak of this distribution moves only marginally during the experiment, to about $5 \cdot 10^{-2}$ cm² after four hours. However, during this shift, a second, smaller mode emerges out of the bulk of the distribution. This second mode corresponds to the larger droplets of the system; those that contribute almost all of the droplet area (cf. fig. 43). The second mode continually diverges, while its peak moves towards larger droplet sizes. Again, peak movement and divergence vary between data series. Generally, the peak after four hours lies between 10 cm² and 100 cm², with the exception of the yellow (20150228) data series, where it lies at just 1 cm².

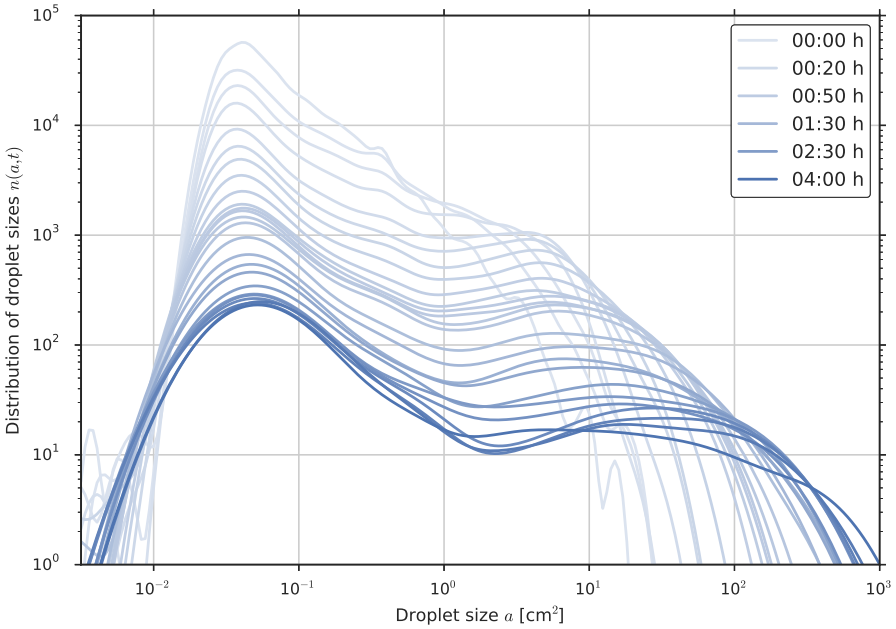


Figure 42: Exemplary number distribution of droplet sizes. The double-logarithmic scales point out the bimodal nature of the droplet size distribution. A first mode with steady peak (in terms of droplet size, and at the very late stages also in its droplet count) of very small droplets is accompanied by a second mode with ever-growing droplet size peak and distribution width.

marginally to the right during the course of the experiment. However, we clearly saw in the previous sections that very large droplets start to form with time. In fact, the distribution becomes *bimodal*. The second mode of this system, initially overlapping the small droplet mode, continually moves its peak to larger droplet sizes. At the same time, it becomes wider and wider, until it is so spread out that it is barely distinguishable in the probability density function.

The bimodal nature of the droplet size distribution becomes more clearly visible if we plot the distribution of droplet sizes, $n(a,t) = N(t) \cdot p(a,t)$ on a double-logarithmic scale, as in fig. 42. Note that this is more than a mere normalisation, as the total number of droplets N varies with time. The declining number of droplets in the experiment corresponds to the declining area under the curve in the number distribution. The first, stable (in terms of peak position) mode of the distribution is clearly visible on the left side, whereas the second mode can be found initially hidden in the right flank of the first mode, then moving towards bigger droplet sizes while spreading out. The peak heights of the two modes are always separated by at least a factor of ten in favour of the small droplets.

The distributions in figs. 41 and 42 seem unintuitive when remembering our raw experiment images from fig. 19. What meets the eye is not the relative distribution of droplet sizes, but the contribution of droplets of size a to the total droplet area, $a \cdot n(a,t)$. This contribu-

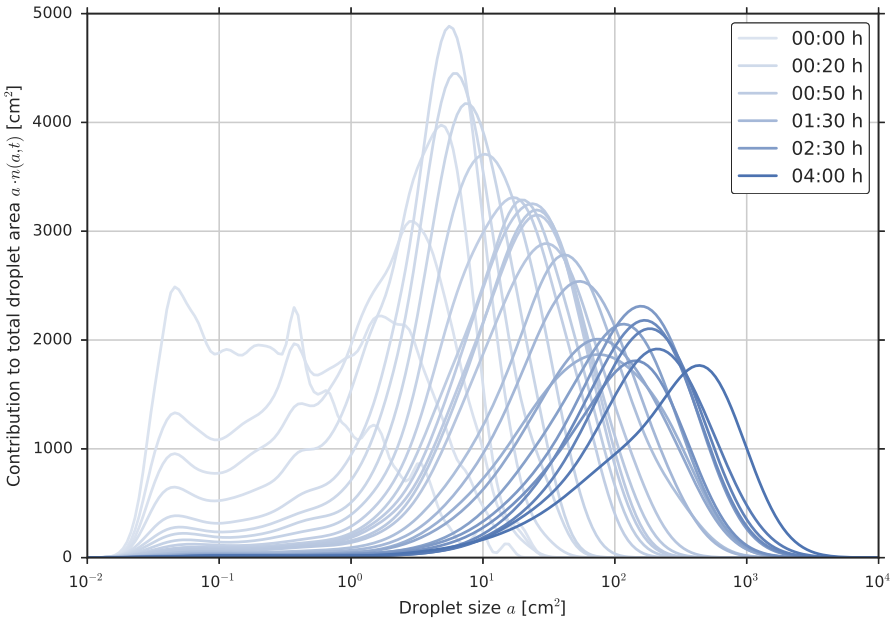


Figure 43: Exemplary distribution of total droplet area. After the first thirty minutes, we find a single-mode distribution with an elongated tail towards smaller droplet sizes. The bulk of the total area is contributed by droplets of sizes 5 cm² to 10 cm². As familiar from fig. 41, the peak continuously moves towards larger droplet sizes while the distribution diverges. After four hours, the largest part of the total area comes from droplets of sizes between 20 cm² and 1,000 cm².

tion is plotted in fig. 43. While not far from negligible in the probability density function in fig. 41, the second mode of the distribution, i. e. the one containing the large droplets, expectedly contributes almost all of the total droplet area. After a short transient of about thirty minutes, the contribution distribution consists of a single mode with an elongated tail towards smaller droplet sizes. At this time, most of the total area is contributed by droplets of sizes 5 cm² to 10 cm². In the following hours, as already familiar from figs. 41 and 42, the single peak continuously moves towards larger droplet sizes while the distribution spreads out. Four hours into the experiment, the bulk of the total area is contributed by droplets sized between 20 cm² and 1,000 cm².

In this section, we found that the distribution of droplet sizes becomes *bimodal* during the experiment. Almost all of the total area is detected by a continually widening peak of few large droplets. At the same time, almost all of the total number of droplets is contributed by a different, narrow and relatively stable peak of small droplets, which take part in none or very few coalescence events.

4.4 TOWARDS A MATHEMATICAL DESCRIPTION

Previously, we saw that the distribution of droplet sizes is bimodal. A first mode of small droplets contributes the bulk of the total number of droplets, while a second mode of large droplets contributes the bulk of the total droplet area. In this section, we investigate these two modes separately and establish that the experiment processes through three time regimes. Finally, we propose and test a breakdown of the distribution of droplet sizes into a superposition of two distributions: a stationary distribution of small droplets, and a scaling distribution of large droplets.

After visual inspection of fig. 42, we can roughly separate the two modes by categorising droplets as either smaller or larger than 1 cm^2. In the following, we will limit us to the '20150218' data set. Figure 44 shows the number of droplets as well as the average and total droplet area for both droplet categories. Marked by grey dashed lines, we can identify three regimes in time:

(i) $t \leqslant 200$ s. Immediately after the experiment's start, when there are very few large droplets, the total detected area is mostly dominated by small droplets. At the same time, we see an increase in number of large, and a decrease in number of small droplets. The average small droplet size, expected to be constant from fig. 42, undergoes a small increase. This can be attributed to an imperfect mode separation by our 1 cm^2 threshold (again, cf. fig. 42).

(ii) $200\text{ s} < t \leqslant 1500$ s. After the experiment has somewhat evolved from its initial conditions, we start to identify several power laws within the data. While the average small droplet size is constant by now, we can see the first manifestation of the moving large-droplets peak in the average large droplet size. It follows a $t^{1/3}$ power law. At the same time, as the total area starts to be dominated by large droplets, the total area contributed by small droplets declines as $t^{-2/3}$. As the average small droplet area is constant, this also implies a $t^{-2/3}$ decline of the number of small droplets. Simultaneously, the number of large droplets stays nearly constant.

(iii) $t > 1500$ s. The third regime comprises around 90 % of the experiment time. Here, the total detected area is contributed almost exclusively by large droplets, and hence the decline in total droplet area manifests in a $t^{-1/3}$ power law. As the coalescence process is now in full swing, their number declines fast, following a $t^{-5/3}$ power law. Accordingly, the average large droplet size scales as $t^{4/3}$. While the average small droplet size remains constant, the total number of small droplets, and consequentially also their total area, undergo a fast decline, losing

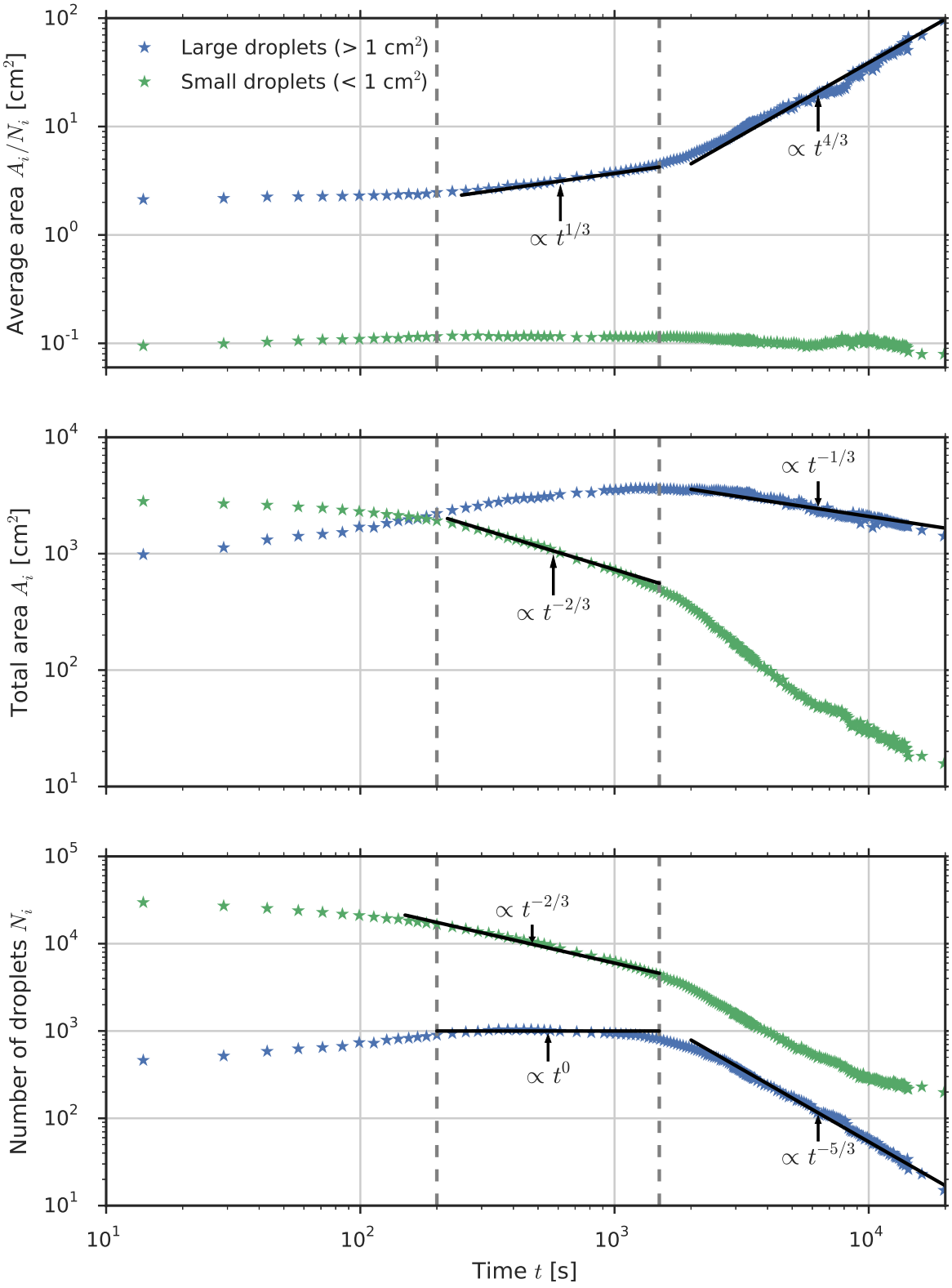


Figure 44: Aggregate statistics per droplet size regime. When building separate aggregates for droplets smaller (green) and larger (blue) than 1 cm², we can identify three time regimes, marked by grey dashed lines. First, the system is dominated by small droplets; then, as the second mode of the droplet size distribution starts separating, larger droplets take over; finally, for about 90 % of the experiment time, both the total area and the number of droplets decline as the coalescence process takes place. Within the aggregate statistics, we can identify several power laws, mostly bound to single time regimes. None of these power laws hold for more than a decade on the time scale. Data shown is from the '20150218' run.

about 90 % of their value in the third regime. This decline flattens out towards the end of the experiment.

Figure 44 also helps in explaining the counter-intuitive behaviour of the (aggregate) average droplet size (fig. 34). While the total area A in $a_{avg} = A/N$ is almost exclusively contributed by large droplets, the total number of droplets N is dominated by small droplets. The aggregate average area mixes up the two modes that make up the droplet size distribution, i. e. it mixes droplets that do with droplets that do not participate in the coalescence process, and therefore has little informative value.

We now suppose a mathematical description of the droplet distribution $n(a,t)$. Based on our findings in the previous sections, we propose that the aggregate distribution is a superposition of

(i) a stationary probability distribution of small droplets, $p_s(a)$, with varying weight, i. e. number of small droplets $N_s(t)$, and
(ii) a scaling probability distribution of large droplets, $p_l(a/\alpha(t))$, dependent on the average large droplet size $\alpha(t) = A_l(t)/N_l(t)$, and a time-dependent weight $N_l(t)$,

such that:

$$n(a,t) = N_s(t) \cdot p_s(a) + \frac{N_l(t)}{\alpha(t)} \cdot p_l\left(\frac{a}{\alpha(t)}\right), \tag{7}$$

$$\text{with} \int_0^\infty da\; p_s(a) = 1,$$
$$\int_0^\infty da\; \frac{1}{\alpha(t)} \cdot p_l\left(\frac{a}{\alpha(t)}\right) = 1,$$

$$\text{i. e.} \int_0^\infty da\; n(a,t) = N(t) \quad \forall\, t.$$

We can check this description by considering the probability $P_>$ that any droplet is larger than a threshold value a^*:

$$\begin{aligned} P_>(a^*,t) &= p(a > a^*) \\ &= \frac{N_>(a^*,t)}{N(t)} \\ &= \frac{1}{N_s(t) + N_l(t)} \int_{a^*}^\infty da\; n(a,t) \end{aligned}$$

At this point, we insert our proposed $n(a,t)$ (eq. (7)). We will now restrain ourselves to large threshold areas a^*. In that case, the con-

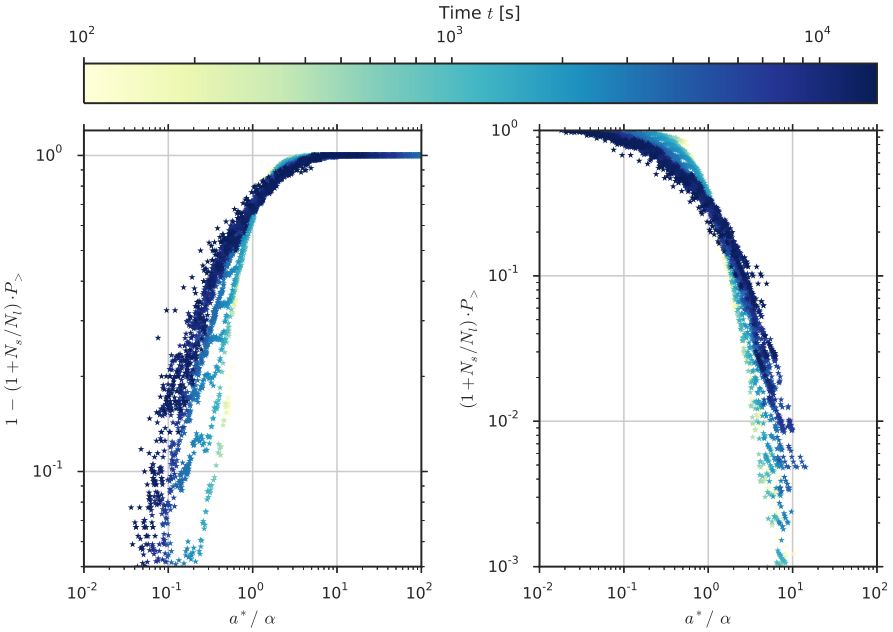


Figure 45: Plausibility check for proposed mathematical description of droplet size distribution (eq. (7)). In particular during the late time regime (blue to dark blue), the plotted values pair fall onto a single line that can be interpreted as the complementary cumulative distribution function for large droplets (cf. eq. (8)). Shown data is composed from samples for 30 different threshold areas a^* between 1.2 cm² and 1000 cm².

tribution of the first summand of eq. (7) to the integral is negligible, and the above can be simplified to

$$\begin{aligned} P_>(a^*, t) &= \frac{1}{N_s(t) + N_l(t)} \int_{a^*}^{\infty} da\, \frac{N_l(t)}{\alpha(t)} \cdot p_l\left(\frac{a}{\alpha(t)}\right) \\ &= \frac{N_l(t)}{N_s(t) + N_l(t)} \int_{\frac{a^*}{\alpha(t)}}^{\infty} dx\, p_l(x), \end{aligned}$$

where the integral can be identified as the complementary cumulative distribution function for large droplets, $\mathfrak{C}_l^>(a^*/\alpha(t)) = p_l(a > a^*/\alpha(t))$:

$$P_>(a^*, t) = \frac{N_l(t)}{N_s(t) + N_l(t)}\, \mathfrak{C}_l^>(a^*/\alpha(t)) \tag{8}$$

As all variables in eq. (8) except for $\mathfrak{C}_l^>$ are experimentally accessible, this allows us to gain a better insight into whether eq. (7) is a plausible description of the distribution of droplets sizes. If it is, all value pairs of $(1 + N_s/N_l)\, P_>$ against a^*/α should be samples of the complementary cumulative distribution function of the underlying scaling distribution of large droplets $p_l(a/\alpha)$, i. e. they should fall onto a single line describing that complementary cumulative distribution function.

Figure 45 shows scatter plots of the above value pairs for a variety of threshold areas. Indeed, we see that the data falls onto a line closely resembling a cumulative distribution function, as we expected from eq. (8). This is true especially for the late time regime, where the separation of the two modes is large and the approximations made to arrive at eq. (8), i. e. that the first summand of eq. (7) is negligible in the integral over large droplet sizes, are applicable. This supports our approach of separating the total droplet size distribution into a stationary distribution for small droplets and a scaling distribution for large droplets.

5 CONCLUSION

The efforts described in this thesis are threefold: Chapter 2 describes a novel physical experiment which grants access to a new level of statistics on the evolution of droplet size distributions during coalescence processes. In chapter 3, I introduced two software frameworks that are designed to be usable beyond this thesis: one facilitates high-speed image capturing at full resolution from consumer-level DSLR cameras, the other aids experimental physicists in the task of organising their image processing. Chapter 4 provides a general evaluation of the experimental data. Unlike what has been published in the literature, the droplet size distributions become bimodal. Hence, in section 4.4 the droplet distribution was modeled in terms of a superposition of two modes: a stationary distribution of small droplets, and a scaling distribution of large droplets.

5.1 ACHIEVEMENTS

EXPERIMENT We use the dynamics of oil droplets on a water surface, familiar from cooking soups, to gain statistics on the droplet size distributions in coalescence processes. The setup is distinctly different from previous coalescence experiments, which either concentrated on single coalescences on µm scales, imaged via microscopy or x-ray methods, or droplet size distributions at a single point in time, such as experiments with fuel injection systems. In our experiment, the detected droplet cover areas from 0.01 cm^2 to 1,000 cm^2, and the experiment initially holds more 20,000 initial droplets, coalescing into a few hundred within four to six hours. A major challenge for the experiment was the visualisation of the droplets, and it has been overcome by exploiting the optical properties of the oil droplets: we take pictures of the shadows that they cast on a screen made of transparent paper when they are illuminated by an LED flashlight.

IMAGE CAPTURING To track large numbers of droplets we used high-resolution digital single-lens reflex cameras, which are widely spread on the consumer markets. However, the available remote control software limits the (full-resolution) frame rate for time lapses to roughly one image every three seconds. For our purposes, a higher frame rate was desirable to avoid losing too much detail of the coalescence process, in particular during the first few minutes of the experiments where the dynamics are very fast. To fill this gap I developed a new open-source software framework, *ICE*, aimed at the needs of

experimentalists. It wraps a low-level camera remote control library and allows full-resolution frame rates up to four frames per second (depending on camera model) by saving images to an SD card instead of immediately transferring them through the camera's USB connection. Thus, it avoids the slow USB bus chips used in consumer-level cameras.

DROPLET DETECTION We automatically detected the oil droplets from the experiment images through a combination of computer vision methods. We find small droplets by utilising their circular shape. A commonly adopted approach based on the Circle Hough Transform, was computationally too intensive due to our wide radii range. We detect circles by finding gradient vector pairs, as proposed by Rad et al.[65]. Moreover, for larger droplets with less well-defined shapes, we use a mixture of thresholding, mathematical morphology, and filtering. As byproduct of implementing the image analysis, I presented a second open-source software framework, *FIRE*, that aims at aiding experimentalists in organising their image processing.

TOTAL DROPLET AREA The five considered, identically prepared, realisations of the experiment diverge quantitatively, and, in some aspects, even qualitatively. This forbids calculating ensemble averages, and strongly suggests the need for more experimental data. Still, many surprising conclusions can be drawn from the present data. Through examination of the area difference between a merged droplet and its two parent droplets, we established that the oil droplets in our experiment are pancake-shaped, i. e. uniform in height except at the edges. Oil mass does not move into the third dimension (height) under coalescences. Still, the total area of detected droplets is not constant. Instead, there is a growing *background* made of oil that does not take part in the coalescence process. We propose that, driven by a decrease in line tension, oil from the center of the system moves into this background through a thin oil layer covering the water.

BIMODAL SIZE DISTRIBUTION After a brief transient, the number distribution of droplet sizes becomes *bimodal*: A first, steady mode of small droplets of constant size contributes the bulk of the total number of droplets. A second mode comprises growing droplets. It contributes the bulk of the total droplet area. By dividing the detected droplets into a group of small and a group of large droplets, we identify three temporal regimes in the evolution. They are characterised by distinct power laws describing the average droplet area, total area, and number of droplets for both groups.

MATHEMATICAL DESCRIPTION Following these observations, a mathematical description for the time evolution of the droplet size

distribution is proposed, where the total droplet distribution is a superposition of a stationary distribution of small droplets, while the size distribution of large droplets takes a scaling form. We support this description with a master plot where the (originally drifting) cumulative distribution function for large droplets falls onto a single line for all but the very early data from an experiment run.

5.2 IMPROVEMENTS

Naturally, no new experiment is born perfect. There are many rough edges in the oil droplet experiment. Many of these can be smoothed easily with a little time, e. g. :

The accuracy and precision of the droplet size detection can be improved by correcting distortions of the camera lens, as commonly done in photo processing software, and by providing a more homogeneous illumination. This can be achieved by buying or building a new light source with a more powerful LED and a wider opening angle, bringing the illumination at the corners more in line with the illumination at the center.

Increasing reproducibility of the experiment, such that the data can be averaged to find ensemble means, can be done by obtaining better control on the initial conditions of the experiment. Right now, we can generate a roughly monodisperse distribution of droplet sizes, but we have no method to define the typical droplet size of this distribution, and little means to set the spatial composition of the initial droplets. This can be achieved by using an atomiser that allows fine control over the deposited mass.

Introducing better control over the surface area is necessary to take advantage of finite size scaling methods. Currently, the amount of oil is predetermined by the aquarium size and our requirements towards the coalescence speed, and there is no tool to change the water's surface area. This can be overcome by either implementing a floating tube as described in section 2.2.4 (with the added benefit of being able to vary the area during a run), or by introducing physical barriers and thus decreasing the effective aquarium size.

5.3 PROSPECTS

So far, we have done little exploration of the experiment's parameter space. For example, we have chosen the oil and the surfactant, and their amounts, based on their availability and a trial-and-error process to find a combination that worked. More extensive and quantitative testing with other mixtures will help identify which of the features we find in our oil droplet experiment are specific to it, and which could broaden our general understanding of coalescence processes.

There are many other open questions in the evaluation of the experimental data. Why do the runs vary in some qualitative features, albeit being prepared in a similar fashion? What are the details of the mechanism through which oil is constantly moved into an inactive background? What laws govern the arrangement of droplets into neighbourhoods with similar droplet size? A particular appeal comes from the bimodal droplet size distributions, a quite uncommon characteristic in the world of statistical physics in general, and a behaviour not yet reported in studies on the coalescence of droplets and bubbles.

Once one can collect reproducible data for a range of different systems, the experiment can be used to explore how droplet size distributions and their evolution are impacted by features of single coalescences, such as by coalescence preference or coalescence cascades. The experiment is a model system that allows full access to the formation of bimodal size distributions, one of the major open problems to address upon modeling rain initiation. Moreover, the superb resolution of the tails of the distributions will make it possible to systematically study the physical significance of features which are usually hidden in those noisy tails, such as the evolution of the largest droplets and its connection to explosive percolation.

A

SOURCE CODES

At the time of writing this, the source codes of ICE and FIRE have a combined length of more than 3,000 lines. Reprinting them here would be of little use. Their source code and documentation are available on GitHub:

- ICE: https://github.com/jdemaeyer/ice
- FIRE: https://github.com/jdemaeyer/fire

The very first commit in both of these repositories corresponds to the state as used in the image analysis of this thesis.

Listing 1: YAML configuration file for image processing with FIRE, as used in the final image processing for our experiment images.

```
preprocsteps:
  - crop:
      y: 252
      x: 942
      dy: 2370
      dx: 2370
  - makefloat
  - greyscale:
      channel: 0
  - removebg:
      bgsource: /run/media/jakob/TOSHIBA EXT/Data/Big
          /20150218/Full/100NCD90/DSC_0006.JPG
      methods:
        - sub
        - div

procsteps:
  # Very large clusters
  - threshold:
      threshold: -0.08
      minsize: 10000
      steps_prelabel:
        - erode: 2
        - remove_bordering: 10
        - fill
        - erode: 2
        - open
        - close
      steps_postlabel:
        - maxstddev: 0.1
        - dilate: 5
        - minconvexratio: 0.8
  # Clusters of all sizes that we are certain about
  - threshold:
```

```
      threshold: -0.05
      minsize: 200
      steps_prelabel:
        - erode
        - remove_bordering: 10
        - fill
        - open
        - calccornerresponse
      steps_postlabel:
        - dilate: 2
        - erode: 2
        - maxstddev: 0.1
        - maxcornerresponse: 7.
        - minQ: 0.3
        - dilate: 2
        - minconvexratio: 0.85
  # Midsized clusters (out of our FCD range) that weren't
      detected previously
  - threshold:
      threshold: -0.03
      minsize: 500
      maxsize: 10000
      steps_prelabel:
        - remove_bordering: 10
        - fill
        - open
        - calccornerresponse
      steps_postlabel:
        - maxstddev: 0.1
        - maxcornerresponse: 6.5
        - minQ: 0.5
        - dilate
        - minconvexratio: 0.9
  - removeouter:
      padding: 80
  - tiledfcd:
      tiles: 5
      minr: null
      maxr: 30.
      gaussian: 1.
      sobel: false
      minnorm: 3.0e-2
      alpha: 0.5
      beta: 0.7
      gamma: 0.3
      mincenterlevel: .1
      radiusscaler: null
      minmembers: null
      epsilon: 1.0
      minsamples: 1
      maxangspread: 0.6
  - diskextend

exportsteps:
  - string
  - save
```

LIST OF FIGURES

LISTINGS

BIBLIOGRAPHY

[1] RS Barnes and DJ Mazey. The migration and coalescence of inert gas bubbles in metals. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 275, pages 47–57. The Royal Society, 1963.

[2] Mauricio Terrones, Humberto Terrones, F Banhart, J-C Charlier, and PM Ajayan. Coalescence of single-walled carbon nanotubes. *Science*, 288(5469):1226–1229, 2000.

[3] Frank H Shu, Fred C Adams, and Susana Lizano. Star formation in molecular clouds - observation and theory. *Annual Review of Astronomy and Astrophysics*, 25:23–81, 1987.

[4] Volker Springel, Tiziana Di Matteo, and Lars Hernquist. Modelling feedback from stars and black holes in galaxy mergers. *Monthly Notices of the Royal Astronomical Society*, 361(3):776–794, 2005.

[5] Bruce G Elmegreen, Frédéric Bournaud, and Debra Meloy Elmegreen. Bulge formation by the coalescence of giant clumps in primordial disk galaxies. *The Astrophysical Journal*, 688(1):67, 2008.

[6] EG Bowen. The fomation of rain by coalescence. *Australian Journal of Chemistry*, 3(2):193–213, 1950.

[7] A Kovetz and B Olund. The effect of coalescence and condensation on rain formation in a cloud of finite vertical extent. *Journal of the Atmospheric Sciences*, 26(5):1060–1065, 1969.

[8] Bjorn Stevens and Graham Feingold. Untangling aerosol effects on clouds and precipitation in a buffered system. *Nature*, 461 (7264):607–613, 2009.

[9] Robert A Sohn, Claire Willis, Susan Humphris, Timothy M Shank, Hanumant Singh, Henrietta N Edmonds, Clayton Kunz, Ulf Hedman, Elisabeth Helmke, Michael Jakuba, et al. Explosive volcanism on the ultraslow-spreading gakkel ridge, arctic ocean. *Nature*, 453(7199):1236–1238, 2008.

[10] Michael Manga and HA Stone. Interactions between bubbles in magmas and lavas: effects of bubble deformation. *Journal of Volcanology and Geothermal Research*, 63(3):267–279, 1994.

[11] Katharine V Cashman and R Stephen J Sparks. How volcanoes work: A 25 year perspective. *Geological Society of America Bulletin*, 125(5-6):664–690, 2013.

[12] LA Utracki and ZH Shi. Development of polymer blend morphology during compounding in a twin-screw extruder. part I: Droplet dispersion and coalescence - a review. *Polymer Engineering & Science*, 32(24):1824–1833, 1992.

[13] Uttandaraman Sundararaj and CW Macosko. Drop breakup and coalescence in polymer blends: the effects of concentration and compatibilization. *Macromolecules*, 28(8):2647–2657, 1995.

[14] DT Wasan, JJ McNamara, SM Shah, K Sampath, and N Aderangi. The role of coalescence phenomena and interfacial rheological properties in enhanced oil recovery: an overview. *Journal of Rheology (1978-present)*, 23(2):181–207, 1979.

[15] Rouhollah Farajzadeh, Alexey Andrianov, Rumen Krastev, GJ Hirasaki, and William Richard Rossen. Foam–oil interaction in porous media: Implications for foam assisted enhanced oil recovery. *Advances in colloid and interface science*, 183:1–13, 2012.

[16] J Qian and CK Law. Regimes of coalescence and separation in droplet collision. *Journal of Fluid Mechanics*, 331:59–80, 1997.

[17] Melissa Orme. Experiments on droplet collisions, bounce, coalescence and disruption. *Progress in Energy and Combustion Science*, 23(1):65–79, 1997.

[18] Pierre-Gilles De Gennes, Françoise Brochard-Wyart, and David Quéré. *Capillarity and wetting phenomena: drops, bubbles, pearls, waves*. Springer Science & Business Media, 2013.

[19] ST Thoroddsen and K Takehara. The coalescence cascade of a drop. *Physics of Fluids (1994-present)*, 12(6):1265–1267, 2000.

[20] François Blanchette and Terry P Bigioni. Partial coalescence of drops at liquid interfaces. *Nature Physics*, 2(4):254–257, 2006.

[21] Hartmut Gau and Stephan Herminghaus. Ripening of ordered breath figures. *Physical review letters*, 84(18):4156, 2000.

[22] Tobias Lapp. *Evolution of Droplet Distributions in Hydrodynamic Systems*. PhD thesis, Georg-August-Universität Göttingen, 2011.

[23] Byung Mook Weon and Jung Ho Je. Coalescence preference depends on size inequality. *Phys. Rev. Lett.*, 108:224501, May 2012.

[24] Yeseul Kim, Su Jin Lim, Bopil Gim, and Byung Mook Weon. Coalescence preference in densely packed microbubbles. *Sci. Rep.*, 5:–, January 2015.

[25] Geoffrey Grimmett. *Percolation*. Springer, Berlin, 1999.

[26] SR Broadbent and JM Hammersley. Percolation processes. *Mathematical Proceedings of the Cambridge Philosophical Society*, 53:629–641, July 1957.

[27] Dimitris Achlioptas, Raissa M. D'Souza, and Joel Spencer. Explosive percolation in random networks. *Science*, 323(5920):1453–1455, 2009.

[28] RA da Costa, SN Dorogovtsev, AV Goltsev, and JFF Mendes. Explosive percolation transition is actually continuous. *Physical Review Letters*, 105:255701, December 2010.

[29] RA da Costa, SN Dorogovtsev, AV Goltsev, and JFF Mendes. Critical exponents of the explosive percolation transition. *Physical Review E*, 89(4):042148, 2014.

[30] Jan Nagler, Anna Levina, and Marc Timme. Impact of single links in competitive percolation. *Nature Physics*, 7(3):265–270, March 2011.

[31] Oliver Riordan and Lutz Warnke. Explosive percolation is continuous. *Science*, 333(6040):322–324, 2011.

[32] Oliver Riordan and Lutz Warnke. Achlioptas process phase transitions are continuous. *The Annals of Applied Probability*, 22 (4):1450–1464, August 2012.

[33] ST Thoroddsen, TG Etoh, K Takehara, and N Ootsuka. On the coalescence speed of bubbles. *Physics of Fluids (1994-present)*, 17 (7):071703, 2005.

[34] FH Zhang and ST Thoroddsen. Satellite generation during bubble coalescence. *Physics of Fluids (1994-present)*, 20(2):022104, 2008.

[35] NA Kazakis, AA Mouza, and SV Paras. Coalescence during bubble formation at two neighbouring pores: An experimental study in microscopic scale. *Chemical Engineering Science*, 63(21): 5160–5178, 2008.

[36] Ivan U Vakarelski, Rogerio Manica, Xiaosong Tang, Sean J O'Shea, Geoffrey W Stevens, Franz Grieser, Raymond R Dagastine, and Derek YC Chan. Dynamic interactions between microbubbles in water. *Proceedings of the National Academy of Sciences*, 107(25):11177–11182, 2010.

[37] Emmanuel Villermaux. Fragmentation. *Annu. Rev. Fluid Mech.*, 39:419–446, 2007.

[38] Jens Eggers and Emmanuel Villermaux. Physics of liquid jets. *Reports on progress in physics*, 71(3):036601, 2008.

[39] RA Mugele and HD Evans. Droplet size distribution in sprays. *Industrial & Engineering Chemistry*, 43(6):1317–1324, 1951.

[40] Hiroyuki Hiroyasu and Toshikazu Kadota. Fuel droplet size distribution in diesel combustion chamber. *Bulletin of JSME*, 19(135): 1064–1072, 1976.

[41] SE Burns, S Yiacoumi, and C Tsouris. Microbubble generation for environmental and industrial separations. *Separation and Purification Technology*, 11(3):221–232, 1997.

[42] Alexander B Kostinski and Raymond A Shaw. Fluctuations and luck in droplet growth by coalescence. *Bulletin of the American Meteorological Society*, 86(2):235–244, 2005.

[43] Douglas R. Hofstadter. *Godel, Escher, Bach: An Eternal Golden Braid*. Basic Books, Inc., New York, NY, USA, 1979. ISBN 0465026850.

[44] Irving Langmuir. Oil lenses on water and the nature of monomolecular expanded films. *The Journal of Chemical Physics*, 1(11):756–776, 1933.

[45] William D Garrett and William R Barger. Factors affecting the use of monomolecular surface films to control oil pollution on water. *Environmental Science & Technology*, 4(2):123–127, 1970.

[46] Robert Aveyard, John H Clint, Dieter Nees, and Vesselin Paunov. Size-dependent lens angles for small oil lenses on water. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 146(1):95–111, 1999.

[47] Emanuel Bertrand, Daniel Bonn, Daniel Broseta, H Dobbs, JO Indekeu, J Meunier, K Ragil, and N Shahidzadeh. Wetting of alkanes on water. *Journal of Petroleum Science and Engineering*, 33 (1):217–222, 2002.

[48] Lena Happ, Lutz Künneke, Jan Lebert, Philip-Michael Materna, and Han Wool Song. Competitive percolation. Available on request, 2011.

[49] Béla Bollobás, Svante Janson, and Oliver Riordan. The phase transition in inhomogeneous random graphs. *Random Structures & Algorithms*, 31(1):3–122, 2007.

[50] Nagwa Ghoneim. Photophysics of nile red in solution: steady state spectroscopy. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 56(5):1003–1010, 2000.

[51] Eugene Hecht and A Zajac. *Optics*. Addison Wesley, New York, 2002.

[52] P Paseiro Losada, C Pérez Lamela, MF López Fabal, P Sanmartín Fenollera, and J Simal Lozano. Two rp-hplc sensitive methods to quantify and identify bisphenol a diglycidyl ether and its hydrolysis products. 1. european union aqueous food simulants. *Journal of Agricultural and Food Chemistry*, 45(9):3493–3500, 1997.

[53] Frank Luther Mott. *A history of American magazines: 1885-1905*, volume 4. Harvard University Press, 1957.

[54] Marcus Meissner, Lutz Müller, Scott Fritzinger, and Hubert Figuiere. libgphoto2. https://github.com/gphoto/libgphoto2/, 2015.

[55] SD Association. SD Standard overview: Speed class. https://www.sdcard.org/developers/overview/speed_class/, 2006.

[56] Ramesh Jain, Rangachar Kasturi, and Brian G Schunck. *Machine vision*, volume 5. McGraw-Hill New York, 1995.

[57] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[58] G. Bradski. OpenCV. *Dr. Dobb's Journal of Software Tools*, 2000.

[59] Kurt Demaagd, Anthony Oliver, Nathan Oostendorp, and Katherine Scott. *Practical Computer Vision with SimpleCV: The Simple Way to Make Technology See*. " O'Reilly Media, Inc.", 2012.

[60] Luis Pedro Coelho. Mahotas: Open source software for scriptable computer vision. *Journal of Open Research Software*, 1, July 2013.

[61] Stefan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, Francois Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[63] C. Sommer, C. Strähle, U. Köthe, and F. A. Hamprecht. ilastik: Interactive learning and segmentation toolkit. In *Eighth IEEE International Symposium on Biomedical Imaging (ISBI 2011). Proceedings*, pages 230–233, 2011. 1.

[64] Richard O Duda and Peter E Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[65] Ali Ajdari Rad, Karim Faez, and Navid Qaragozlou. Fast circle detection using gradient pair vectors. In *DICTA*, pages 879–888. Citeseer, 2003.

[66] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[67] John Illingworth and Josef Kittler. A survey of the Hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116, 1988.

[68] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654.

[69] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6:679–698, 1986.

[70] HK Yuen, John Princen, John Illingworth, and Josef Kittler. Comparative study of Hough transform methods for circle finding. *Image and vision computing*, 8(1):71–77, 1990.

[71] Carolyn Kimme, Dana Ballard, and Jack Sklansky. Finding circles by an array of accumulators. *Communications of the ACM*, 18(2):120–122, 1975.

[72] Stefan van der Walt, S. Chris Colbert, and Gael Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[73] Vladimir Estivill-Castro. Why so many clustering algorithms: a position paper. *ACM SIGKDD explorations newsletter*, 4(1):65–75, 2002.

[74] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[75] Nicholas I Fisher. *Statistical analysis of circular data*. Cambridge University Press, 1995.

[76] Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 6:583–598, 1991.

[77] Georges Matheron. *Eléments pour une théorie des milieux poreux*. Masson, 1967.

[78] JC Klein and J Serra. The texture analyser. *Journal of Microscopy*, 95(2):349–356, 1972.

[79] Pierre Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[80] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2015-09-19].

[81] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.

[82] Jing Li and Nigel M Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10): 1771–1787, 2008.

[83] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.

[84] Hans P Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, DTIC Document, 1980.

[85] Robert Osserman. The isoperimetric inequality. *Bulletin of the American Mathematical Society*, 84(6):1182–1238, 1978.

[86] Jack Sklansky. Measuring concavity on mosaic. *IEEE Transactions on Computers*, 1972.

[87] CF Mora and AKH Kwan. Sphericity, shape factor, and convexity measurement of coarse aggregate for concrete using digital image processing. *Cement and concrete research*, 30(3):351–358, 2000.

[88] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[89] James Henstridge, Johan Dahlin, Tomeu Vizoso, Martin Pitt, Paolo Borelli, Ignacio Casal Quinteiro, Sebastian Pölsterl, and Simon Feltman. PyGObject. `https://wiki.gnome.org/PyGObject`, 2015.

ERKLÄRUNG

nach §17(4e) und §17(9) der Prüfungs- und Studienordnung für den Studiengang Physik an der Georg-August-Universität Göttingen:

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüberhinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, im Rahmen einer nichtbestandenen Prüfung an dieser oder einer anderen Hochschule eingereicht wurde.

Göttingen, 30 September 2015.

Jakob de Maeyer