Universität Göttingen
Fakultät für Physik

Bachelorarbeit am
Max-Planck-Institut für Dynamik und Selbstorganisation

Analysis of the motility modes of Shewanella Oneidensis MR-1

Thomas Breithaupt

thomas.breithaupt@stud.uni-goettingen.de

Matrikelnr.: 21215734

July 16, 2018

Supervisor and first referee: Dr. Marco G. Mazza
Second referee: Prof. Dr. Jörg Enderlein

Max-Planck-Institute for
Dynamics and Self-Organization

Department Dynamics of
Complex Fluids

Nonequilibrium
Soft Matter

Contents

1 Introduction

Shewanella oneidensis MR-1 is a Gram-negative, rod-shaped bacterium with one flagellum as propeller. Its movement is guided by stimuli, such as chemical gradients in its environment. Typically $2 - 3\mu$m long and $0.4 - 0.70\mu$m in diameter it can be mainly found in both marine and limnic environments in nature. *Shewanella oneidensis* belongs to a class of bacteria called *exoelectrogenic* because it has the capacity to transfer electrons to its environment. The electron transfer mechanisms implemented by *Shewanella oneidensis* are part of its respiratory cycle. Because of its electrical activity, it can reduce poisonous heavy metal ions like uranium and live in both anaerobic and aerobic environments. *Shewanella oneidensis* has the ability to produce electrical conductive nanowires up to three times their own cell length, connect to each other and form large biofilms. The collective behavior due to chemotaxis and cell-cell interaction is not fully understood yet. Analyzing the movement towards complex collective patterns is the objective of this work and is studied by means of particle-tracking algorithms. As there is no general solution to track *Shewanella oneidensis*, the focus is on the tracking algorithm itself.

2 Microswimmer Dynamics

2.1 Microswimmers

Microorganisms like bacteria range in size from $1\,\mu$m to $100\,\mu$m, they can move with velocities up to $1\,$mm/s. The velocity decreases with increasing body size. Small bacteria like *Shewanella* with a size of about $3\,\mu$m can move with up to $100\,\mu$m/s [1]. This velocity is about 33 times their own body size per second. If we compare it to a human of about $2\,$m it corresponds to $66\,\text{m/s} = 237\,\text{km/h}$ which is vastly greater than the maximum speed of a human. Large microswimmers like Paramecium are $100\,\mu$m large and move with only up to 10 times their own body size, $1\,$mm/s.

Microswimmers in a fluid like water obey Navier-Stokes equations. Active swimmers like bacteria generate a flow field around them. The fluid set into motion by one microswimmer will perturb the motion of another nearby microswimmer. The motion of the latter will in turn perturb the motion of the former. These effective interactions, mediated by the surrounding fluid, are called hydrodynamic interactions. Although they are generally important, they will not be the focus of our work.

2.2 Low Reynolds Numbers

Many microswimmers have a nearly buoyant density. That means their density is similar to their environment and they freely float in the water. Hence gravity has close to no effect on them, regardless if they are moving or not. Due to the small size of bacteria we need to examine the acting forces on the bacteria and their surroundings. In response to the force exerted by the bacterial flagellum, the water exerts hydrodynamic forces on the moving bacteria. Osborne Reynolds considered the ratio of the inertial forces and the viscous forces given by

$$R = \frac{\text{inertial forces}}{\text{viscous forces}} = \frac{\rho v L}{\eta} = \frac{vL}{\nu},$$

where v is the velocity, L is the characteristic length, ρ is the density and η is the viscosity or $\nu = \frac{\eta}{\rho}$ is the kinematic viscosity. R is known as the *Reynolds* number. Small values of R indicate that viscous forces dominate over viscous forces, a regime valid for microswimmers. For example, a human swimming in water has a Reynolds number of 10^4. For a small fish the Reynolds number goes down to 10^2. For bacteria we observe a Reynolds number ranging from 10^{-4} to 10^{-5}. Hence, the viscous forces dominate and the inertial force is negligible. If the bacterium stops its own propulsion, it will stop immediately due to the strong viscous forces.

The Navier-Stokes equation for an incompressible Newtonian fluid is

$$-\nabla p + \eta \nabla^2 \boldsymbol{v} = \rho \left(\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\, \boldsymbol{v} \right)$$

For small Reynolds numbers we can neglect the inertial terms and the Stokes equation reduces to

$$\nabla p = \eta \nabla^2 \boldsymbol{v},\ \nabla \cdot \boldsymbol{v} = 0,$$

where the second equation represents the condition of incompressibility of the fluid. For example, a bacterium with a length of $1\,\mu$m and a velocity of $30\,\mu$m/s will stop after $0.01\,$nm or $0.001\,\%$ of its own size after $0.6\,\mu$s. So swimming at low Reynolds numbers is entirely determined by the forces at the present moment and not by the past history.

2.3 Taxis and biofilm formation

Taxis is a common biological behavior characterized by directed movement along a varying external stimulus. This stimulus can be an oxygen gradient, the so-called *aerotaxis*. Other usable elements such as metals are utilized by the bacterium with the *chemotaxis*. The bacterium follows a chemical gradient towards the nutrition

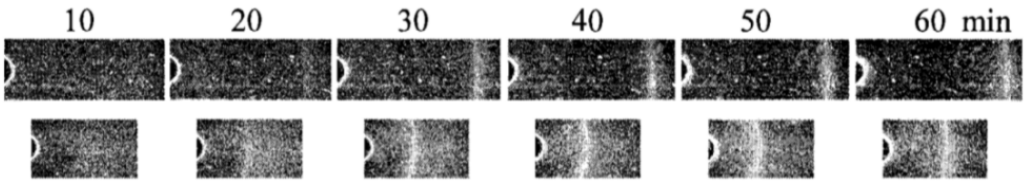


Figure 1: Formation of an aerotactic band and aerotaxis towards air bubble. Reprinted from [2].

source. Proteins on the outer layer of the bacterium are able to measure and compare chemical concentrations at different times and positions which is called *tropotaxis*. This taxis mechanisms finally leads to a dense and motile bacterial band moving towards the nutrition source, as Figure 1 illustrates. Specially in the case of aerotaxis this band is called *aerotactic-band*.

The majority of bacteria is capable to build a cluster of sessile bacteria. Some are also capable to switch from a motile to a sessile phase and form a cluster. If the bacteria reach their desired condition via aerotaxis or chemotaxis, they attach to it and build a network to form a biofilm. In fact all bacteria embedded in a slime layer are called biofilm [3]. This slime is made up of approximately 80% bacteria and a solution of water and agglomerates produced by the bacteria. Biofilm formations are already found 3.25 billion years ago in fossils [4] and seems to be common for 80% of bacteria living on this planet. In nature biofilms of different bacteria strains are found in a variety of habitats from deserts or oceans to the arctic. The preferably inhabit different interfaces [5]. Beside solid surfaces in liquids like rocks in the ocean, interfaces between liquids of different densities like oil in water are also preferred. Biofilms are also observed at gas-water interfaces and solid to gas interfaces when there is sufficient water.

3 Shewanella

Forty different species of the genus *Shewanella* and their habitats have been studied since 1931 [6, 7]. Whereas some live in the ocean or lakes, others inhabit mud all over the world as Figure 2 shows. The *Manganese Reducer* (MR) strain received attention when a significant higher amount of Mn^{2+} than the natural oxidized form Mn^{4+} was found in lake Oneida in New York State [8]. It is also the first *Shewanella* strain whose genome was completely sequenced [9].

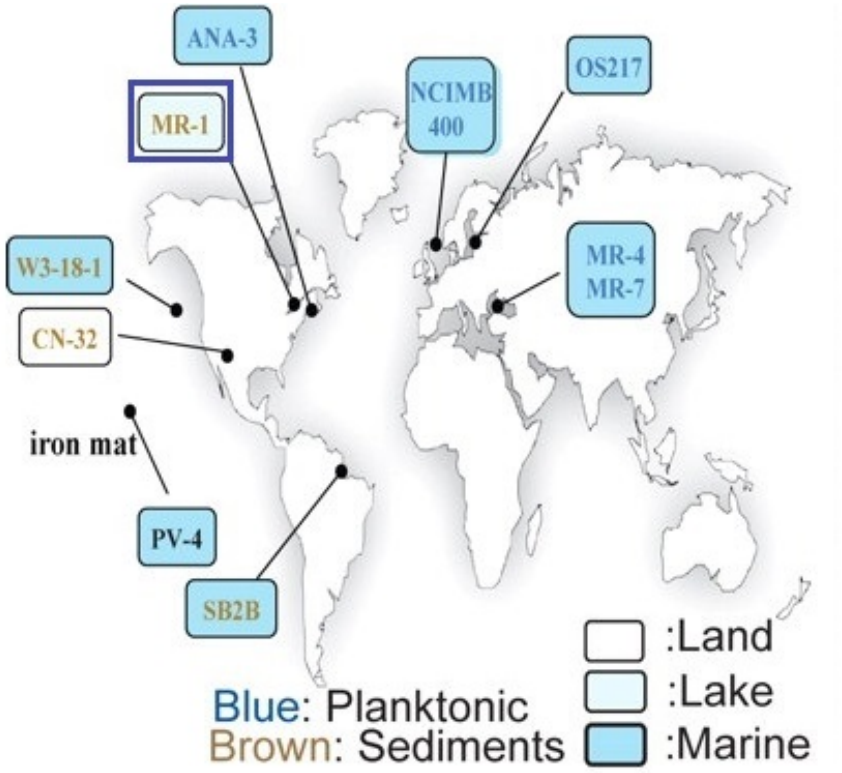


Figure 2: Geographic distribution of the different strains of *Shewanella* that have been isolated. MR-1 strain localized in lake Oneida in New York State highlighted in a blue box. Reprinted from [6].

3.1 Respiration mechanisms

Metabolism in a biological system can be reduced to a respiratory cycle terminating in an electron donation to one's environment. Oxygen is one of the most important electron acceptors on Earth, for example in the respiratory cycle of humans. As opposed to the mitochondria of humans, bacteria evolved numerous different methods for extracellular electron transport (EET). There can be various electron acceptors beside dissolved oxygen in the environment of a bacterium. Insoluble electron acceptors such as metal oxides and protein complexes are used in nature.

3.1.1 Microbial nanowires and extracellular electron transfer

Shewanella oneidensis is capable of producing and transferring electrons through nanowires and electron carriers [10, 11, 12]. Under anaerobic conditions they are more likely to produce nanowires and stop moving to form a biofilm [13, 14]. They use EET for congregation near insoluble electron acceptors such as iron, manganese and uranium [15]. Figure 3 illustrates the EET process. The *Cytochromes OmcA* and *MtrC* are proteins that can transport electrons to particles in direct contact with the bacterium [17]. One important protein used for EET is riboflavin [18].

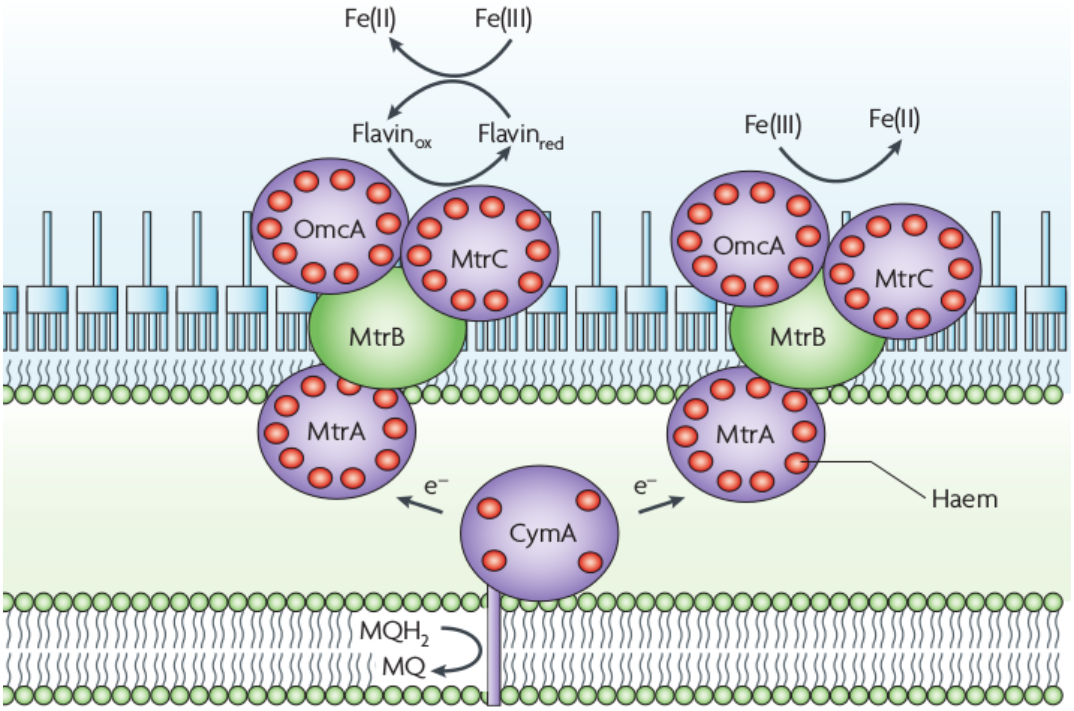


Figure 3: Schematic representation of the electrical membrane with focus on the multiple proteins involved in the transport of electrons. Reprinted from [16].

Riboflavin enables electron transport up to 50 μm from the cell without physical contact. It is also possible to transfer electrons through nanowires because they are an extension of the outer cellular membrane with the necessary proteins for it [11]. The nanowires length ranges from a few nanometers up to a few micrometers. In Ref. [19] they were found to have an average length of 2.5 μm over 100 randomly chosen nanowires and were up to 9 μm long. They transport up to 10^6electrons/s and have an electrical resistivity of around 1Ωcm [10, 20]. After the bacteria congregate and form a biofilm around an electron acceptor they produce nanowires to exchange more electrons. The nanowires allow bacteria to interact with their surroundings.

3.2 Areas of application

Shewanella oneidensis MR-1 is widely used to study anaerobic electron acceptors because their surface is found to be electronegative and hydrophilic [21]. It is a natural way for *Shewanella* to bound to electron acceptors to reduce sulfates, nitrates and chromates beside heavy metal ions.

3.2.1 Radioactive waste

From an environmental point of view, the most interesting application for *Shewanella oneidensis* is the bioremediation of radionuclide contaminated environments. *Shewanella oneidensis* MR-1 has the capability to immobilize solved complexes of radioactive materials [22]. For example, the waste of nuclear power plants contains solved radioactive complexes of uranium which can be immobilized and then separated from non-radioactive waste. As an example, 250 μmol of U(VI) can be reduced to a harmless amount within 48 hours [22]. Plutonium is beside uranium another important reduction partner of *Shewanella oneidensis*.

3.3 Swimming behavior

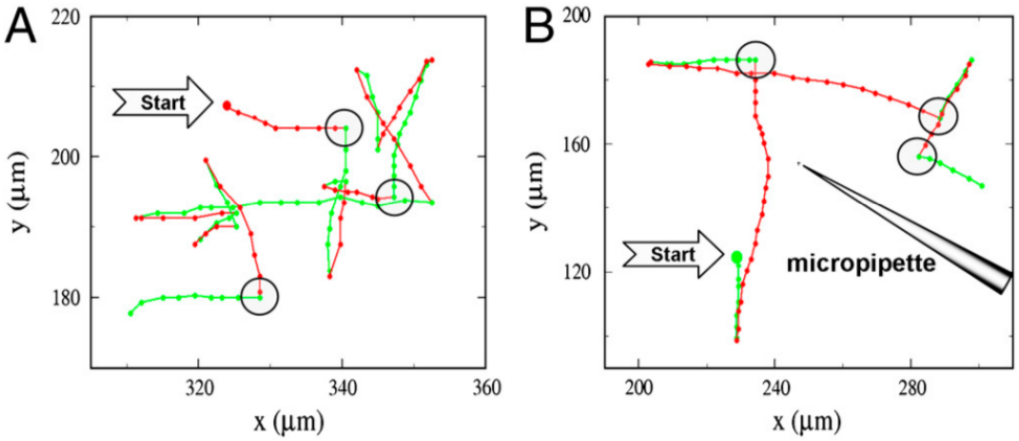


Figure 4: Trajectories of motile bacteria. (A) with no chemical gradient and (B) with a chemical gradient created with a micropipette filled with 1 mmol serin. Green and red trajectories correspond to forwards and backwards swimming, respectively Every dot represents a 0.067 s time step and the big circles were reversal events. Not every reversal is marked. Reprinted from[23].

Shewanella oneidensis has only one flagellum for propulsion. It can rotate counterclockwise and clockwise to swim forwards and backwards, respectively, and it has the ability to drastically change the orientation of the flagellum to change swimming direction, the so-called flick. The direction changes are influenced by a chemical gradient in the surrounding medium. A chemical gradient like an oxygen concentration produced by an air-water interface induces a signal the bacteria want to follow. The moving pattern of the bacterium always follows the same cycle as Figure 4 illustrates. It starts with a forward motion which is then followed by a sudden change

of motion of the flagellar motor that produces backwards motion of the bacterium. As the bacterium changes again from backwards to forwards motion, a mechanical instability at the base of the flagellum produces a sudden reorientation, the flick. The bacterium then resumes its forward motion. The time interval between forward and backward motion is short. For *Vibrio alginolyticus*, also a single-flagellated bacterium, the time between forwards to backwards movements is $\sim 1/30$s, and between backwards and forwards movements is $\sim 1/10$s [23]. So switching from forwards to backwards is 3 times faster than switching from backwards to forwards. The flick of *Vibrio alginolyticus* results in an average direction change of 90°, but is very well distributed over the whole range of 0 to 180. Something similar should be expected for a bacterium with one flagellum like *Shewanella oneidensis*. In contrast to most single-flagellum bacteria *Shewanella oneidensis* MR-1 has two potential propulsion systems [24]. There seems to be a not-negligible number of bacteria with two stator systems in their genome [25]. It is suggested that these stators can switch the motor for the same flagellum in order to use different food sources for propulsion. The mechanism behind the motor regulation is still a subject to study. A proton stator and sodium stator in the *Shewanella oneidensis* MR-1 genome is identified and accessible for movement [24].

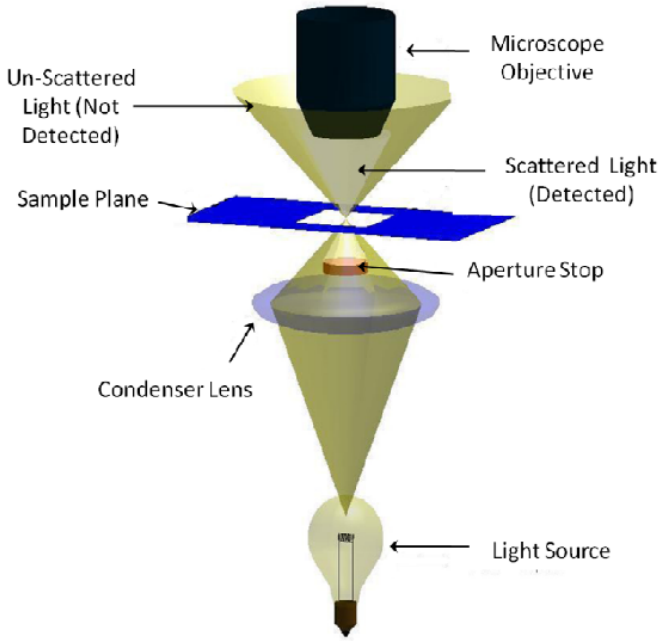


Figure 5: Dark field microscope illumination path scheme. The aperture stop blocks direct illumination to the sample. Only scattered light from the sample reaches microscope objective. [1]

[1] http://physwiki.apps01.yorku.ca/index.php?title=File:Fig2_darkfield.png

4 Particle tracking algorithm

4.1 Preparation

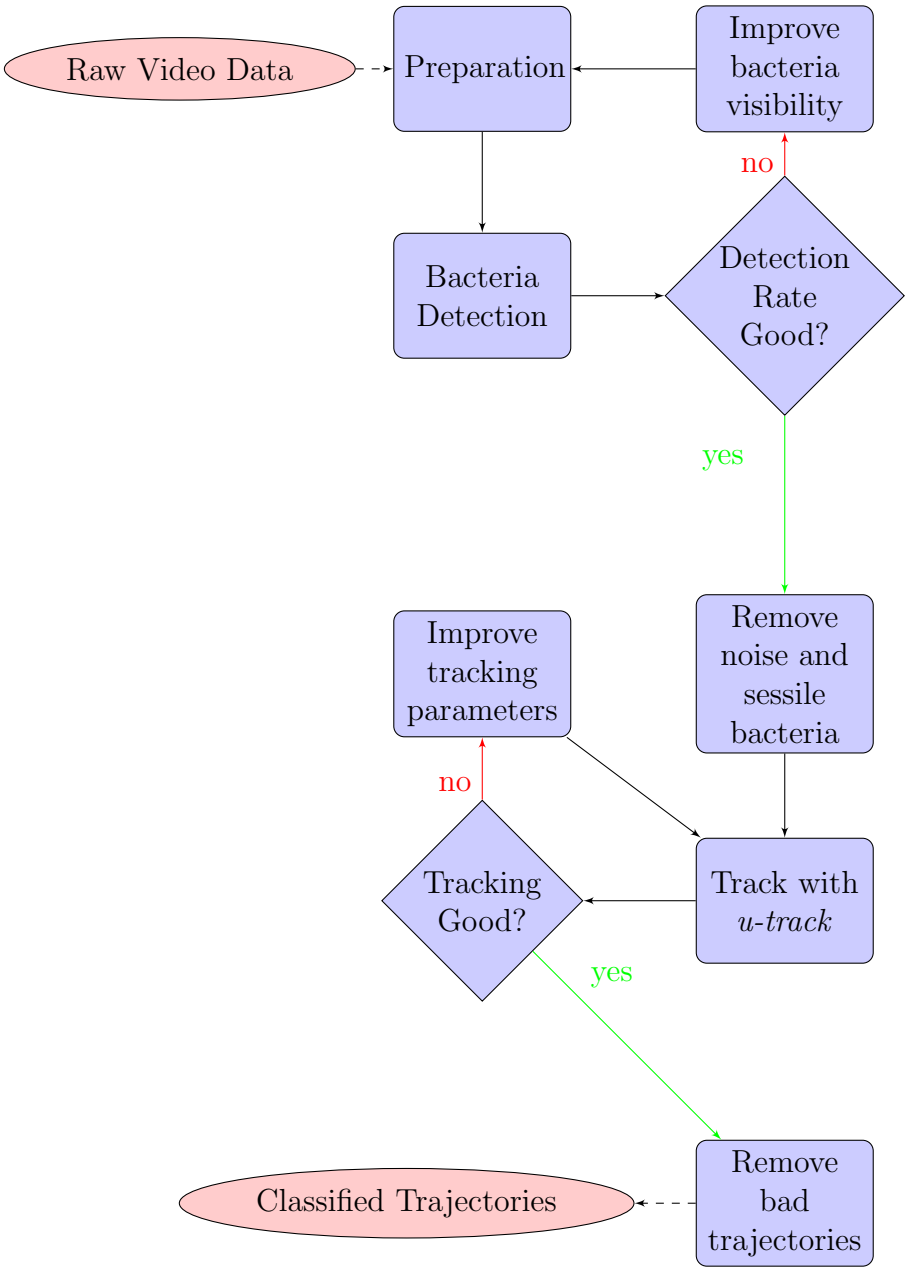


Figure 6: Flowchart of the algorithm for tracking.

Movies of the experiment were recorded at 100 frames per second (FPS) using a dark-field microscope with a camera by Dr. Isabella Guido and her colleagues. A dark-field microscope uses only indirect light to enhance contrast of transparent objects like bacteria. Figure 5 illustrates the illumination path. The light source is focused with a lens to the sample area, but an aperture stop blocks the direct light path to the microscope objective. The sample is only reached by scattered light

which is focused by the microscope objective.

In our experiments we use two lenses one after the other in the objective. The first in the light path ranges from 5x to 63x magnification while the second can be set to 1x, 1.5x or 2x magnification. Therefore the objective magnification ranges from 5x to 126x. The experiment is carried out in a quasi-2D confinement. While the third dimension is at a necessary minimum of around 30 μm, the main area spans about 300 μm $\times$ 300 μm at a 20x2x magnification. Problems with the bacteria motility will arise with smaller confinements, because the provided nutrition is consumed too fast, and therefore the bacterial motility is hindered or suppressed. A trapped bubble within the confinement provides oxygen diffusing into the liquid medium for the bacteria. The camera used in our experiments captures one frame in 1/100s. So it is fast enough to catch the bacterium stopping between frames when switching from backwards to forwards if the timings for *Shewanella oneidensis* MR-1 are similar to *Vibrio alginolyticus*. The microscope can focus only a plane within the confinement, hence the bacteria can swim in and out of focus. At the bottom of the glass confinement, the highest amount of moving bacteria is observed. Also the highest amount of sessile bacteria is found at the bottom, because some tend to stick to the glass or sediment to the ground when they are dead. The focal plane of the videos is fixed at the bottom, nevertheless.

Every image of the recorded video has 2060 pixel $\times$ 2048 pixel at 16bit gray-scale resolution. This gives about 8.5 MB/frame or about 850 MB/s with videos of total duration from 5 to 50 seconds with an uncompressed image format. Fiji [26], a software distribution of ImageJ2 [27] is used in this chapter to analyze and process the data. In addition to a good graphic interface, a headless batch mode for the processing on a computer cluster is integrated. The amount of memory and computational units can be assigned as start arguments. A flowchart of the main steps of the tracking algorithm is shown in Figure 6. Some of the analyzed data consists of 50 s videos. This gives about 42 GB of data per video. As the videos need to be loaded into memory to be processed, they are split into small parts of 3 s, to pre-process them separately.

4.1.1 Image processing

Background noise and fluctuations are unavoidable when recording experimental data. The recorded videos are processed with different filters and tools before the bacteria are tracked. This reduces computational costs for the following algorithms and also improves the result. Figure 7 shows the brightness distribution of a microscope image. The shown intensity interval is already cut down to the first quarter, otherwise the image would just be black. In fact, most of the following images are

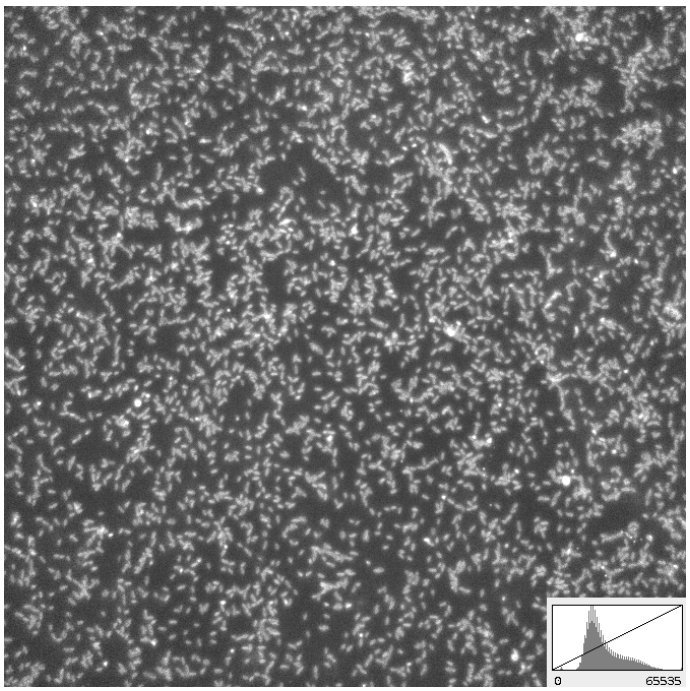
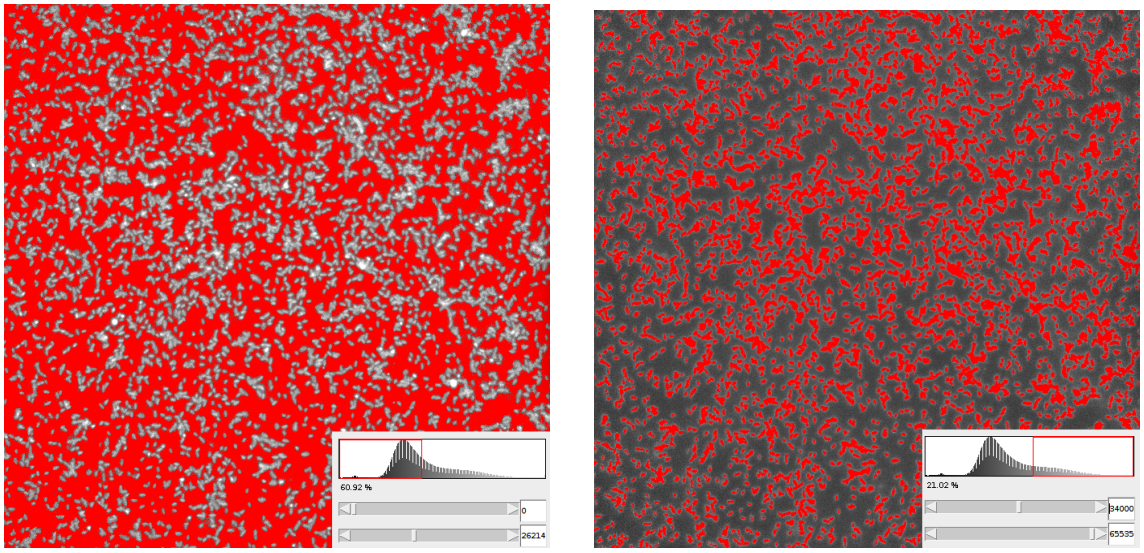


Figure 7: Original image of dark-field microscope at 20x2x magnification. The inset shows the brightness distribution

overexposured to emphasize the issue. The intermediate step detection result in Figure A.1 is almost at the original illumination. The desired final image has a low background noise at a fraction of the bacteria intensity and bacteria around the same intensity. This results in the highest detection rate with the subsequent detection algorithm. The bacteria are clearly visible, but some are brighter than others. The different brightness can be related to different thickness, a different offset with respect to the focal plane, or a different orientation of the body. Figure 8 compares the first peak of the brightness distribution inset with the slowly decaying tail afterwards. The peak identifies as the background. Technically, the bacteria can be identified by selecting the right brightness threshold. Bacteria clusters just appear as one big mass, because the background around them is brighter. Also a fraction of the background is sparkling with the same brightness of the bacteria, seen as little black dots in Figure 8a. These minor problems are overcome with the right detection algorithm, but the main issue is the fluctuation of the brightness over time. In addition to small fluctuations, every 50th image can be significantly brighter than the rest. This results in a high error in the detection phase, as Figure 9 illustrates. It can also happen every 5th frame, for instance. Selecting the right brightness for every image by hand is not feasible. Unfortunately, the background is not always so easy to identify. In some cases the background and the bacteria brightness mixes a lot more. Figure 10 shows such an example. The brightness of the background is



(a) Threshold of background in red

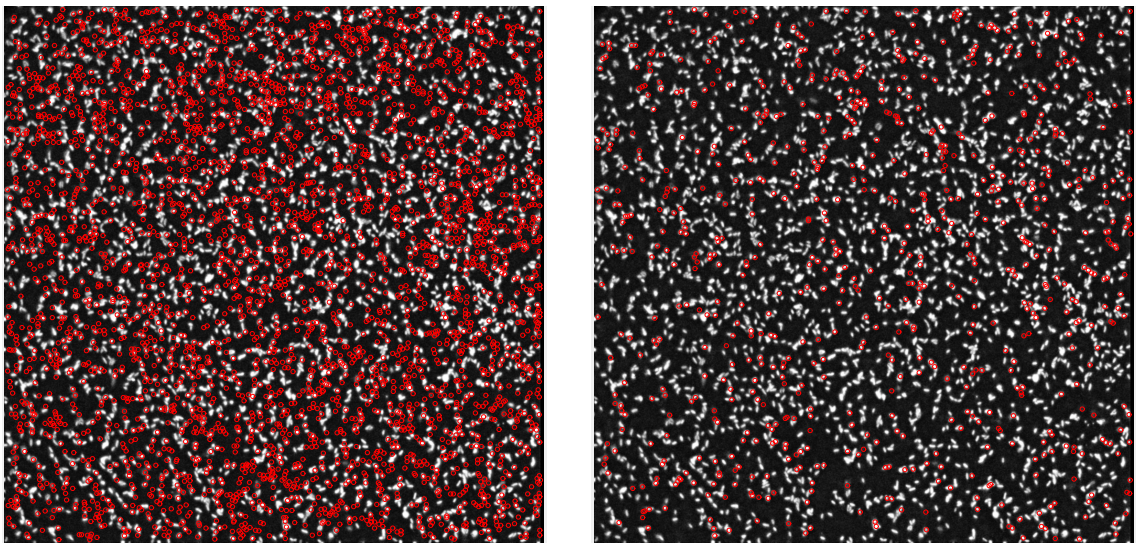
(b) Threshold of bacteria in red

Figure 8: Examples of threshold selection of background and bacteria brightness from Figure 7.

varying considerably in the image and is harder to identify automatically.

There is a rather simple solution for all the issues above, a band-pass filter in Fourierspace. The built-in plugin `FFT/Band-Pass Filter` in `Fiji`, filters objects of a specific size and adjusts the brightness of all objects. Figure A.2 shows an example of the filter. A range of 3 to 40 pixels ($0.48\,\mu\text{m} - 6.46\,\mu\text{m}$) seems to work very well for all experiments at 20x2x magnification. The bacteria brightness is easier to separate from the background now. Here, a useful technique is the ‘rolling background subtraction‘, which removes almost all of the background, as Figure A.3 illustrates. The algorithm ‘rolls‘ over the image with a ball of a specific radius and removes the local average background. Minimal size for the radius is at least the target size. In this case, a radius down to 15 pixels ($2.4\,\mu\text{m}$) leaves the bacteria brightness untouched at 20x2x magnification. It is a little bit lower than the average bacterial size. Doing this after the band-pass filter results in a higher detection rate. Although the rolling background subtraction method removes the background locally and can therefore remove varying background intensities very well, the bacteria are much clearer after the band-pass filter, like Figure A.2b illustrates. Figure A.3 shows a selected area of the subtraction. Large unoccupied areas are not removed, because there are no bacteria to compare the background to. Therefore, the image brightness needs to be adjusted afterwards. After this step the bacteria should be the only thing left in the image. If it is not the case, the parameters needs to be adjusted until this is the case.

The next step is to make the bacteria evenly illuminated. So the geometrical



(a) High detection rate in the background (b) Detection rate in consecutive frame

Figure 9: Detection rate of two consecutive images in one video. Detections at the same position are removed, as described in Section 4.4. Detections of motile particles and background noise remain as red circles.

center can be easier detected. A *Gaussian-Blur* does the trick. The blur is applied with an image convolution [28]. *Fiji* has also a well written plugin for that. A matrix convolution is also used to detect the bacteria later. The input image is weighted with a convolution matrix. For instance the 3×3 matrix

$$\begin{bmatrix} 104 & 100 & 108 \\ 99 & 106 & 98 \\ 95 & 90 & 85 \end{bmatrix} \times \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = 98.\overline{3} \tag{1}$$

takes 9 input pixels, weighs them with the matrix to one output pixel, which is located at the center of the matrix. The image edges are either untouched or interpolated from the surrounding. *Fiji* uses the nearest edge pixel for the out-of-range pixels. The blur is kept low with a kernel size of 1, which corresponds to a 9×9 matrix in a 16bit image, to minimize blurring between touching bacteria. This size almost matches the short axis of the bacteria. Figure A.4 illustrates the difference in bacteria brightness. The bacteria in the top right corner are connected after the blurring, but they can still be detected as different bacteria.

After applying the *Gaussian-Blur* the detection algorithm misses around 36% of the bacteria as Figure A.4 illustrates. The detection result in Figure A.1b shows the highest detection rate, but the detection point is not always in the center of the bacteria. In some cases the brightest point is preferred because it looks like a cell nucleus. This results in an unwanted detection error of the bacteria position. A

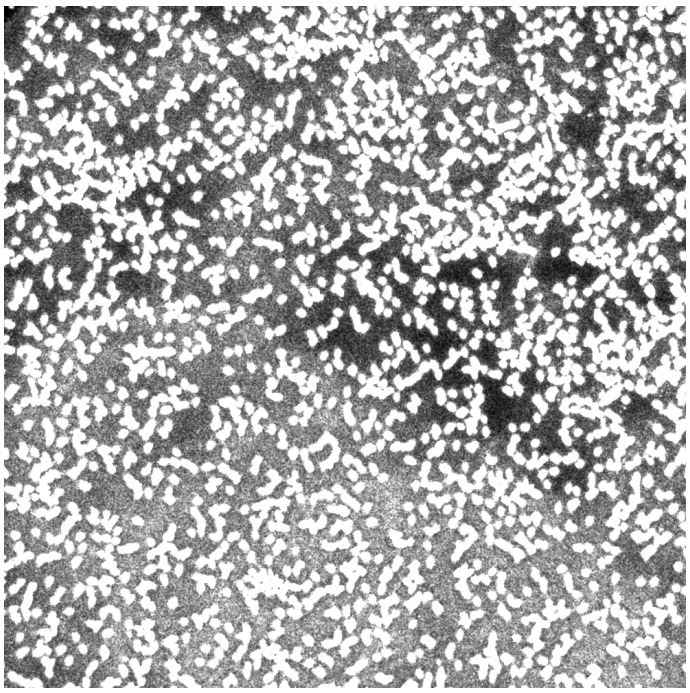


Figure 10: Example of varying background illumination as it occurs in the experimental recordings.

visual check indicates low detection rates around bacteria with less background noise. A test image is converted to a binary gray-scale, where bacteria are represented as ones and the background as zeros. Figure A.1d shows a small area of such an image. The detection rate drops compared to the *Gaussian-Blur* image A.1c. Therefore, the next step is to add artificial background noise. Although a detection algorithm for a detection without background noise is available, it is not used. The built-in detection algorithm in *u-track* is used for convenience and compatibility with the *u-track* tracking algorithm. Before adding noise, the image is converted to 8-bit, to reduce the size of the images. It splits the size exactly to half without any compression method. This means each pixel has values on a gray-scale from 0 to 255. The bacteria intensity is typically in the top 50% if not in the top 90%. The artificial background is made up of a Gaussian distribution with a standard deviation of 5 pixels and a mean of 0. Figure 11 shows the overexposured background together with the bacteria. The detection rate goes up to 95% after all the proposed steps, as Figure 12 illustrates.

4.1.2 Normalization

The experiments are recorded at either 40x4x or 20x2x magnification and overviews at 5x1x or 10x1x. For later analysis we need to convert a pixel to μm. A small μm ruler is placed at all possible magnifications under the microscope. The markers on

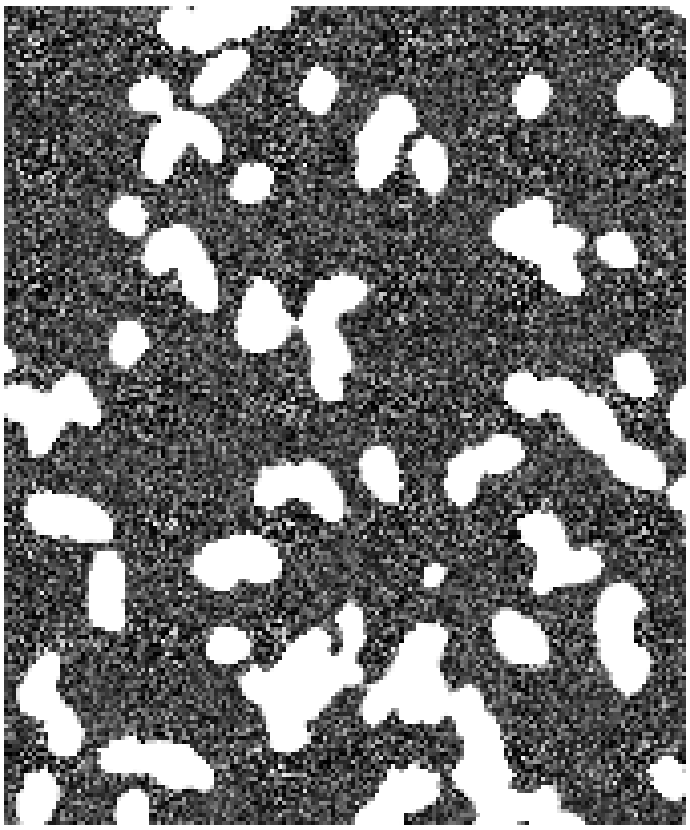


Figure 11: Example of a final image with the addition of artificial background noise; the image is overexposured.

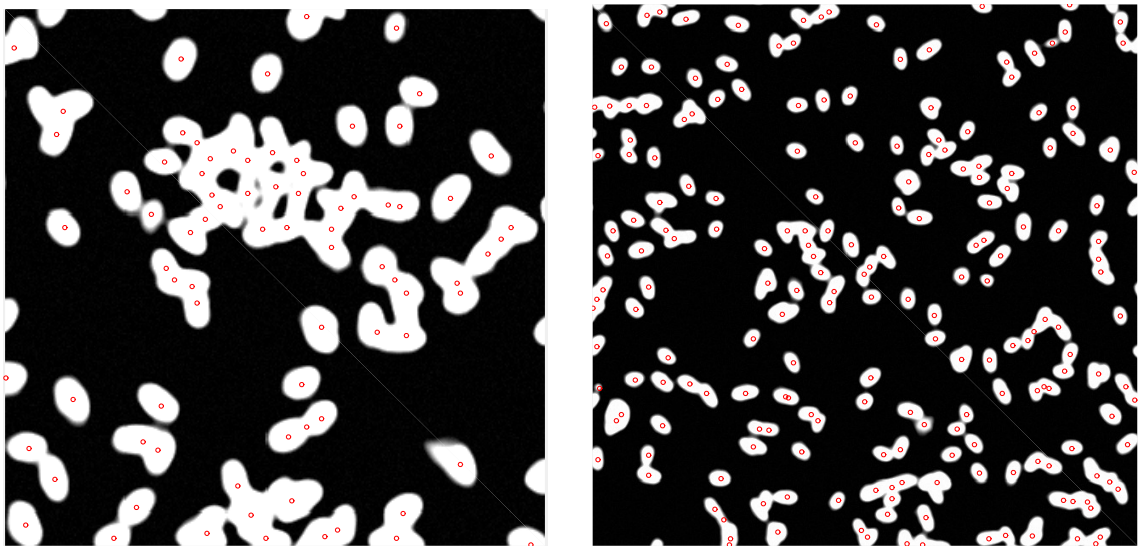
the ruler span over several pixels. Multiple measurements with different markers on the ruler give a good average for the pixel-to-μm conversion. Table 1 shows the results. The primary part of the used data is at 20x2x magnification. With 0.1617(1) μm/pixel and 100frames/s we obtain

$$1\,\frac{\text{pixel}}{\text{frame}} = 16.17\,\frac{\mu\text{m}}{\text{s}}$$

at 20x2x magnification. So a displacement of 1 pixel/frame is already significant. The measured conversion in Table 1 shows that a doubling in magnification, does not exactly double the conversion factor. For instance 40x2x/20x2x = 0.0811/0.1618 = 0.50124 is still above 0.5 within the margin of error. This discrepancy is within the normal manufacture range.

4.1.3 Bacterial size

After the images are processed, the average bacterial size is measured. This size is important for bacteria detection in dense systems, to detect the right amount of bacteria in dense regions where bacteria overlap. Therefore, the processed images



(a) Detection rate in dense bacteria cluster, ∼ 95% correct

(b) Detection rate of a control area, ∼ 95% correct

Figure 12: Test of detection rate result with bacteria cluster and single bacteria.

Magnification	[μm/pixel]
5x1x	1.2789(309)
10x1x	0.6464(76)
20x1x	0.3238(9)
40x1x	0.1612(7)
63x1x	0.1035(2)
5x2x	0.6466(45)
10x2x	0.3226(33)
20x2x	0.1617(1)
40x2x	0.0813(2)
63x2x	0.0517(1)

Table 1: Conversion from pixel to μm for all magnifications

described in Section 4.1 are used to get the average size for this specific analysis method. As *Shewanella oneidensis* MR-1 has a rod-like shape and the orientation is uncertain in a quasi 2D confinement, the average is taken over all shapes. Both the long axis a and short axis b are measured. If there is no clear long and short axis, just one diameter is measured which is then called circular axis. The measured axis is always the diameter because it is easier to identify than the radius from the center. The size is measured by hand with the *line selection tool* in the standard toolbox of Fiji [26]. After selecting a line we use either *Analyze* $\rightarrow$ *Measure* or *CTRL* + *M* and select the distance in the options. All measurements can be easily exported to a simple text file for example. The distribution of measurements is shown in Figure 13 for all axes.

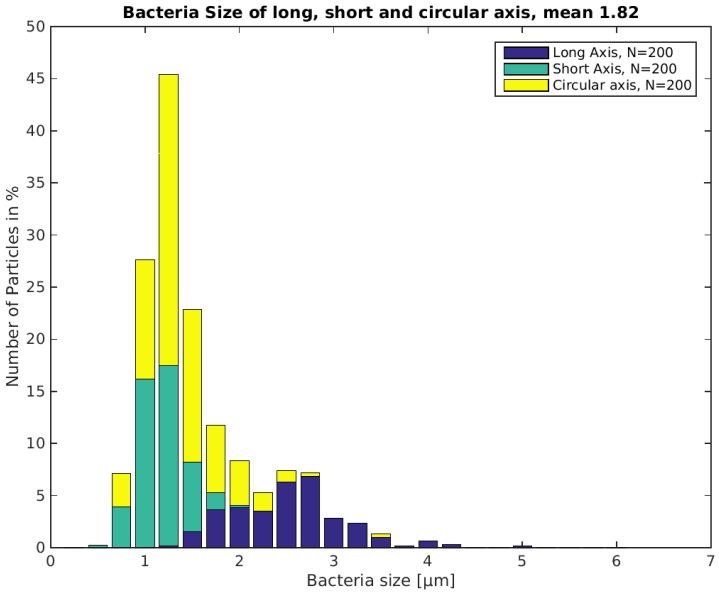


Figure 13: Probability distribution of detected bacterial size. We identify the long, short and circular axis after image processing.

The mean value of 1.82 μm is not representative for the actual average bacterial size, because the measured diameters are not normally distributed. Every distribution itself is normal distributed, but all together are not. Two peaks can be identified corresponding to the small axis and the long axis. The size of the detected circular shapes range from the short axis to the long axis. This can be explained with different orientations perpendicular to the focal plane. Measurements by hand are always faulty, so the short axis is not always perfectly distinguishable from the circular shapes. But a rod-like shape appears brighter in the microscope image, because more volume reflects more light to the camera objective in a dark-field microscope. The higher amount of reflected light could lead to a circular shape. This is supported by the higher number of long axis below their average. The rod-like shape is not fully visible in the focal plane and is therefore shifted to the shorter axis.

4.1.4 Detecting the air bubble

Some videos are recorded with a large white object close to one boundary of the frame. It is the oxygen source for the bacteria and needs to be detected for two reasons. The first reason is to remove all bacteria detections on and nearby the bubble, since the bubble intensity overwhelms the bacteria intensity and causes false detections. These detections needs to be removed before the detection linking between different frames is done. The second reason is that the location of the bubble surface is used to calculate statistical properties of the bacteria, such as

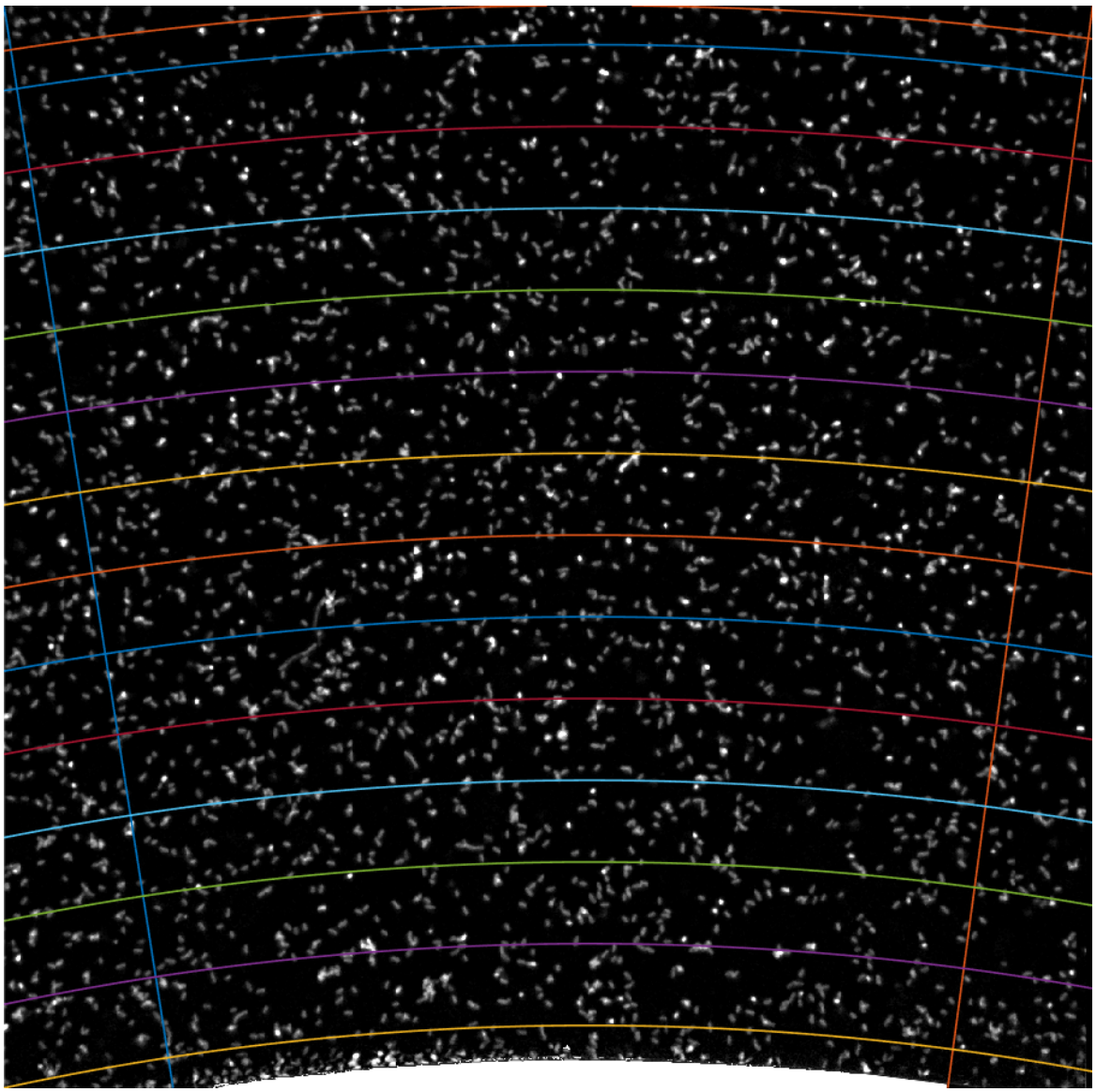


Figure 14: Video split into 13 circular intervals within a fixed circle sector.

distance and the relative angle of the swimming direction to the bubble surface. For simplicity, all images are rotated until the bubble is located at the bottom. Although the reorientation is computed and saved automatically, subsequent calculations and orientations simplify with a bubble at the same position for all images.

A rolling background subtraction removes all objects which are significantly smaller than the largest object. If the resulting image contains a bubble, it will be the only thing remaining. If more than one object remains, the object with the largest area is chosen by counting the white pixels. The boundary of the object is estimated and a circle is fitted to the outer border. The position and radius is saved in a database. The remaining region from bubble surface to the opposite image edge, is divided in intervals for all frames, as Figure 14 illustrates. However, the δ_R

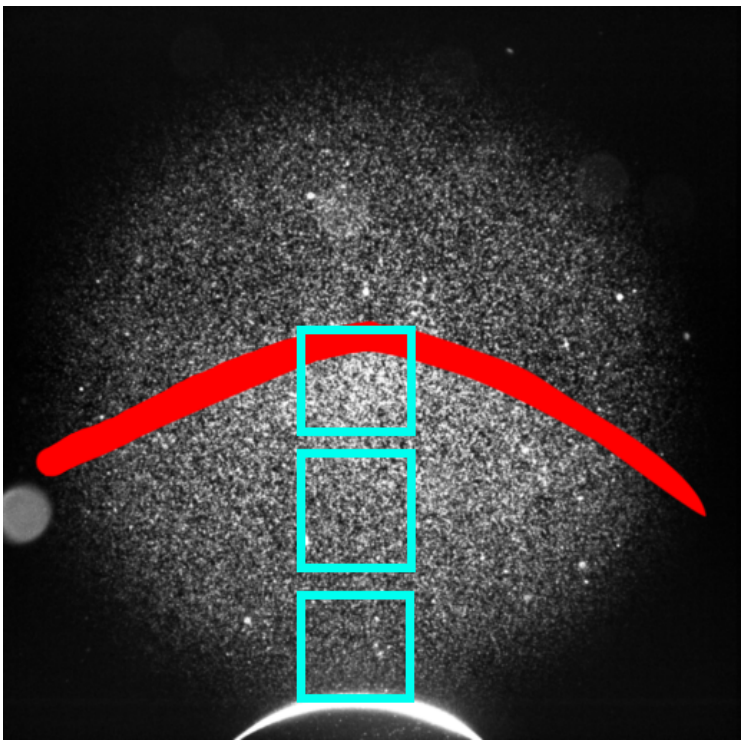
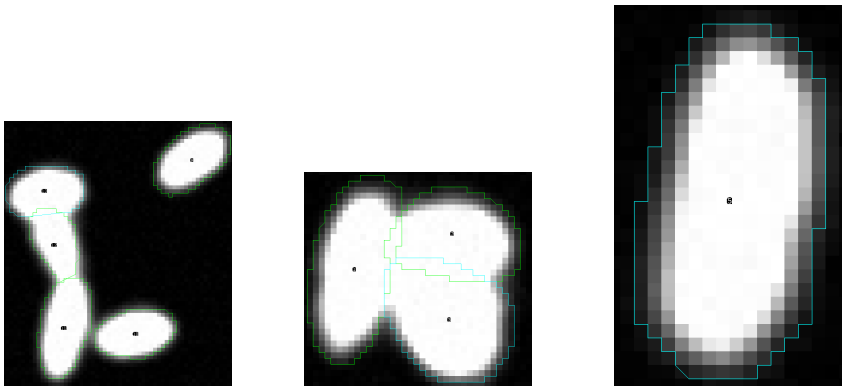


Figure 15: Overview of the experiment showing the air bubble to bacterial band. Blue boxes indicate the locations.

between subsequent intervals is different for different videos. To simplify the calculation of the interval area, a circular sector with fixed angle ϕ is chosen, so that the end of the circular sector matches the image corners opposite to the bubble. Figure 14 shows an example with 13 intervals where the last is significantly smaller. The first interval is set to match one lower image corners with its lower circle in a way that the circle does not leave the image to the site of the bubble. In this way the first circle avoids leaving the image before hitting the circle segment borders. Due to the size of the air bubble, the lost area outside of the intervals is about % with this method. At the used magnification the distance from bubble to the video border is around 300 μm. But the region of interest spans up to 1200 μm and gets smaller for every following video. The region of interest is assigned from the air bubble at the bottom of Figure 15 until the red arc.

Multiple videos are recorded in the region of interest. The distance to the bubble is then assigned according to the location reference within the region of interest. In general the region of interest is bigger than the area of all recorded videos together. Hence equally spaced regions like the blue boxes in Figure 15 illustrate are recorded. The same bubble is assigned to the videos further away from it in the same manner as before. Caging circles leaving the image to the left and right and the boundary of the circle segment hits the image corners as in Figure 14.

4.2 Bacteria detection



(a) Sample of bacteria cluster

(b) Second sample of dense bacteria cluster

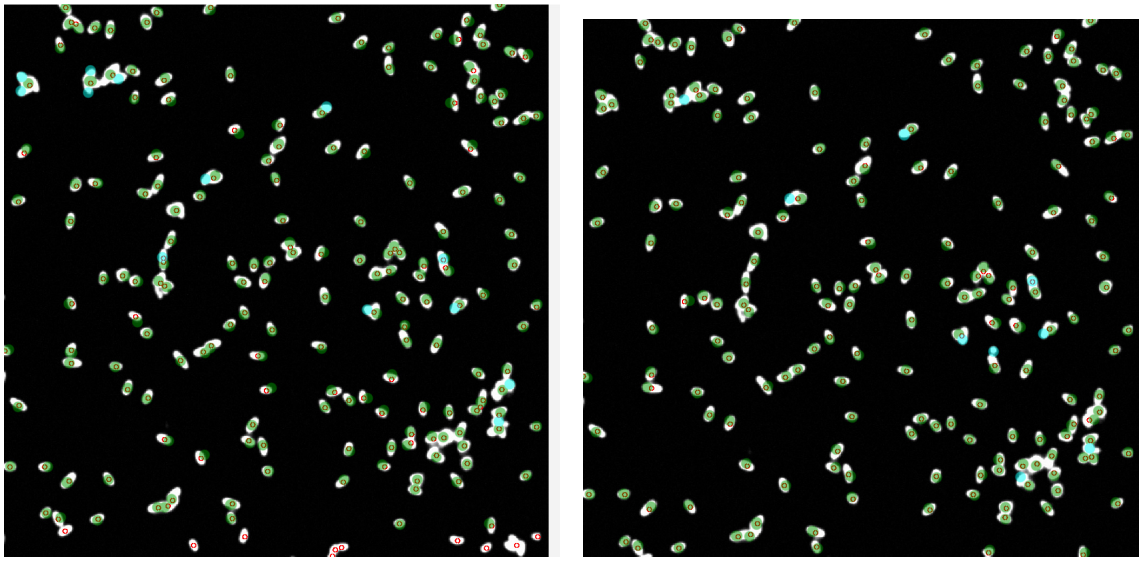
(c) Sample of single bacteria

Figure 16: Sample of bacteria images used to create an artificial data set. (a) and (b) show overlapping and touching bacteria, whereas (c) shows a single bacterium.

The software package *u-track* has 5 different detection algorithms. Only two are useful for bacteria: point source detection, and nuclei detection. Nuclei-detection uses three different mechanisms to distinguish a particles from the background. If one fails, a more complex and robust method is used. The algorithm is designed to detect the nuclei within a cell. But what we want is to track correctly the geometrical center of the bacteria. The resolution and quality of our images is anyway too low to see the actual nuclei. Instead the brightest spot within the bacteria is chosen as the nucleus in a gray-scale image. After image processing as described in Section 4.1.1, most importantly through a Gaussian blur according to Equation 1, the bacteria are evenly illuminated. Hence the geometrical center of the bacteria is detected, like one big nucleus. The only parameter is the nucleus radius, while other parameters are calculated on the fly and depend on the respective data set. Because of this method, some error messages occur during the detection process. However this workaround seems to deliver the highest detection rate of all built-in algorithms within *u-track*.

Compared with nuclei detection, point source detection has 4 additional parameters. One specifies the background and the other 4 criteria for bacteria. Point source detection also takes about 3-5 times longer than nuclei detection. For a more detailed description of the detection algorithms, see the official source code of *u-track*[2]. We tested both algorithms with artificial and real data, to find the optimal solution for our experiments. The artificial data are made up with randomly chosen

[2]https://downloads.openmicroscopy.org/u-track/2.1.3/

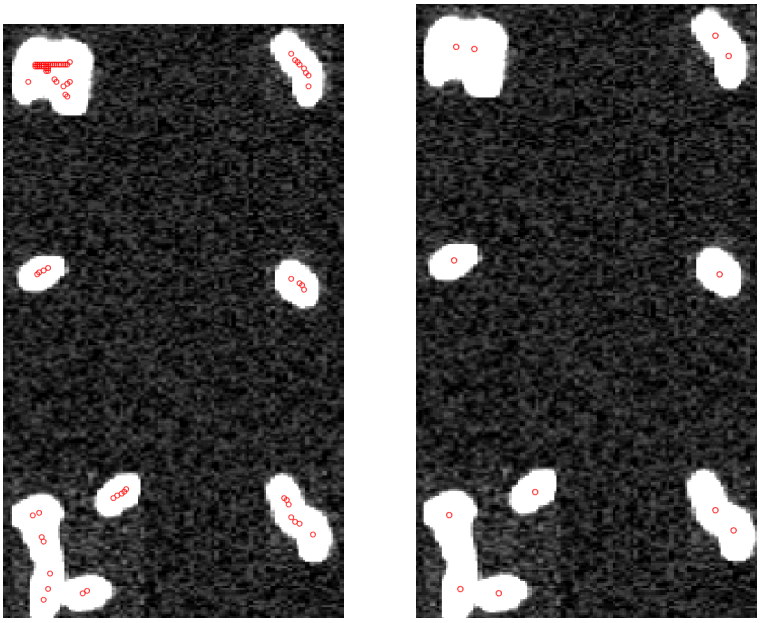


(a) Detection with detecting nuclei (b) Detection with point source detection

Figure 17: Comparison of bacteria detection rate in real data set with nuclei detection (a) and point source detection (b). Detections marked with a red circle; correct detections overlaid with a green circle and false positive or missing detections are overlaid with a blue circle.

real data placed on a black image with random background noise. Bacteria from the real data set are cut out and saved separately as an image. A small sample of 5 single bacteria and 5 clusters as in Figure 16a and 16c is used. In the artificially constructed images, they are always placed with equal spacing, so that they will never overlap or touch the image boundaries. The chosen clusters have bacteria aligned in different ways. Either overlapping, long to long axis or long to short axis. For background noise the *rand()* function in *MATLAB* is used to add a random number between 1 and 10 to every pixel of the image after the bacteria are added, stopping at the limit 255. Because the bacteria are cut out of a real data set, the intensity around the bacteria is higher than in the rest of the image by a small percentage. But this also happens in the real data set and is therefore allowed. As the bacteria are placed artificially, the exact position is known. Their center is determined by hand using *Fiji* $\rightarrow$ *free hand selection tool* and *Measurements tools*. The circles in Figure 16a, 16b and 16c show the selected bacteria with their center as a black dot. Determining the center of touching bacteria is however hard by hand and probably faulty. But the resulting offset between hand and algorithm detection should still be consistent with multiple detections of the exact same bacteria image.

The same artificial image is used for both algorithms that is, nuclei detection and point source detection. For the nuclei detection algorithm a radius of 2 pixels or $0.323\,\mu$m works best to obtain a higher number in clusters in artificial data, whereas



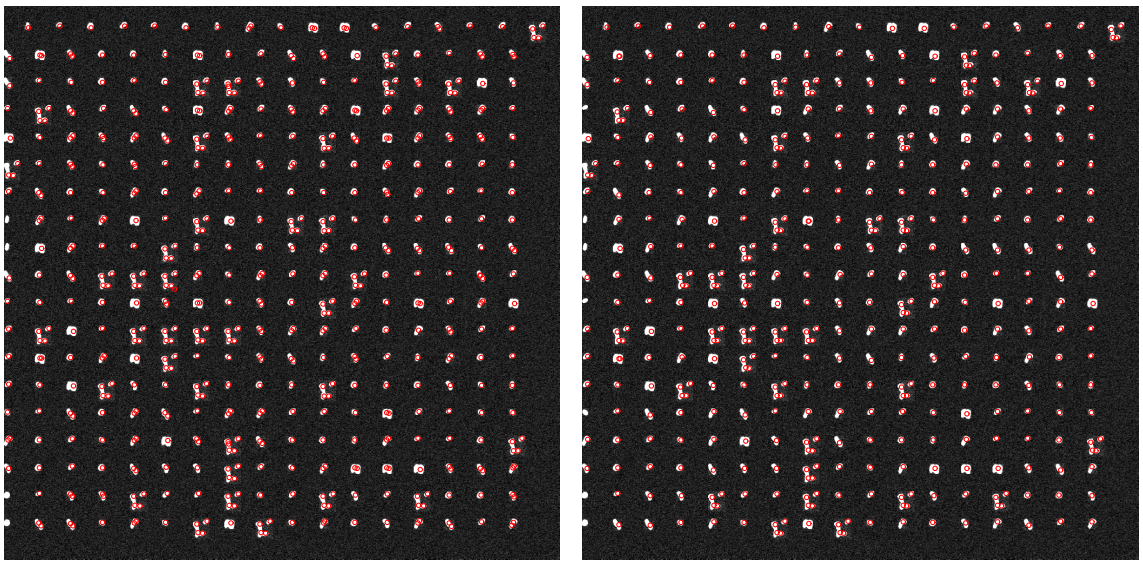
(a) Detection with 1 pixel nuclei radius

(b) Detection with 2 pixel nuclei radius

Figure 18: Comparison of bacteria detection rate with 1 pixel nuclei radius (a), and 2 pixel nuclei radius (b). Detections are marked with a red circle.

the point source detection has a higher rate in some cases in the real data set, see Figure 17. Figure 17 shows the detection rate for both algorithms with overlaid marks for correct and false detections. A nucleus radius of one pixel results in too many detections, see Figure 18. Only integer numbers are allowed, so nothing in between is possible. But in consecutive frames the detection rate varies considerably for both algorithms due to different shapes of clusters, and no clear winner can be found. Taking a closer look at Figure 19 shows different positions and detection numbers for the same cluster from Figure 16a, even without changing the shape. A more dense cluster like in Figure 16b identifies an advantage of the nuclei detection algorithm. It is more likely to detect at least two bacteria, whereas the point source detection only detects one in most cases. None of them is able to detect all 3 bacteria clearly, using different detection parameters for the algorithm. Also the position of the bacteria within the cluster seems to be more clear with the nuclei detection. To verify this, the exact positions are used to calculate an offset from the position measured by hand.

A statistical comparison between both algorithms of the detection rate and position of the artificially placed bacteria is shown in Figure 20. A radius of 30 pixels or about 2 bacterial sizes around a detection point is checked for a correct position. If there is more than one, the closest is always chosen. Assigned detections are deleted in the known positions, so no data points are assigned twice. The detection rate of



(a) Nuclei detection of artificial data

(b) Point source detection of artificial data

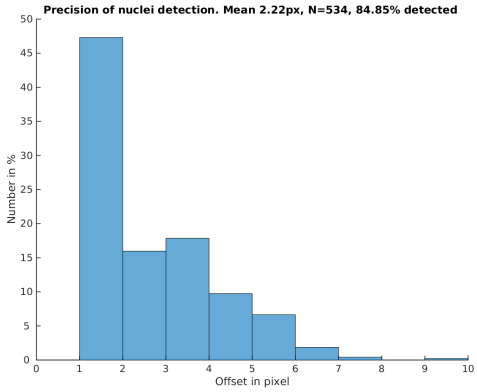
Figure 19: Comparison of bacteria detection rate in artificial data set with nuclei detection (a) and point source detection (b). Detections are marked with a red circle.

the nuclei detection algorithm is $\sim 85\%$ where the point source algorithm only finds $\sim 68\%$. Also the mean position offset in Figure 20b is ~ 1 pixel higher than Figure 20a. Also the offset does not decay as well in the point source detection as in the nuclei detection. A second peak appears in the point source detection at around 7 pixels. These values correspond to the cluster detections and confirms the nuclei detection as the better algorithm again. Also the detection rate at the left image edge in Figure 19 is better with nuclei detection.

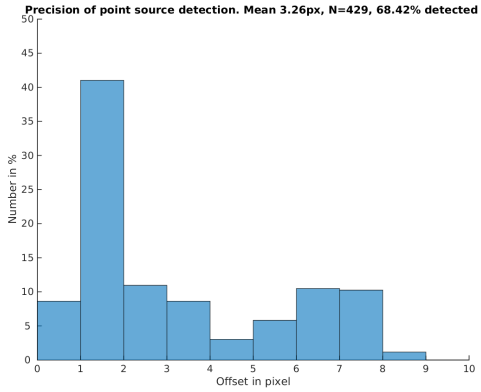
Combining the different detection results, we determine that nuclei detection is the suitable choice. After all the above steps, the detection rate of nuclei detection is around 95% with real data. We determined the detection rate by using a small sample data set like in Figure 12b and 17, where the amount of detected, undetected and wrong detected bacteria is counted by hand. This rate seems high enough to continue with the tracking. A visual check suggests the detection errors occur only in clusters and touching bacteria and is consistent with the artificial data.

4.3 Tracking algorithms for dense systems

The most accurate solution to track single particles is the method of multiple-hypothesis tracking (MHT) [29]. Although we use a simplified, less computationally expensive version is used [30], the basic concept is briefly discussed. In MHT, the particle positions in every frame are used to construct all possible trajectories within



(a) Nuclei detection offset. Detection rate 85%



(b) Point source detection offset. Detection rate 68%

Figure 20: Comparison of bacteria detection offset in artificial data set with nuclei detection (a) and point source detection (b). Detection rate difference of 17% and second peak in (b) are both due to cluster detection issues.

some boundary parameters for each trajectory. The largest non-conflicting ensemble of trajectories is chosen at the end. The solution is then optimal in space and time for all trajectories. Constructing such a large ensemble is very computationally expensive and needs a lot of memory, if not impossible to calculate after some extent of video length or amount of data points per frame. Therefore, a globally optimal solution in time and space needs to be approximated. The most obvious step is to take a local solution. Linking from frame to frame is the lowest temporal solution. But then a bacterium would be lost if it were missing in one frame. To overcome this, the linking also includes a reasonably extended search radius [31]. This greed for finding the next best candidate can of course lead into trouble. Especially for the analyzed data in this work, because the major part of the bacteria does not move (see Section 4.1) they can be preferred during a collision between motile and sessile bacteria with this algorithm. However, bacteria are expected to be visible again after vanishing in a small local area. Identifying the right bacteria after losing track of them, is the biggest barrier in tracking dense systems. A specific method to reduce the number of candidates for the experiments considered in this thesis is described in Section 4.4. Most of the sessile and motile bacteria are identified before the tracking. Linear assignment problem (LAP) [32] is one possible solution to the issues listed above and chosen as the way to go here. The LAP optimizes both linking features by weighting different links and chooses the optimal solution. After linking from frame to frame, generated track segments are linked together. The track linking is globally temporal optimized where the initial frame linking is spatial optimized. But the separated optimization can still conduct to false linking.

4.3.1 u-track

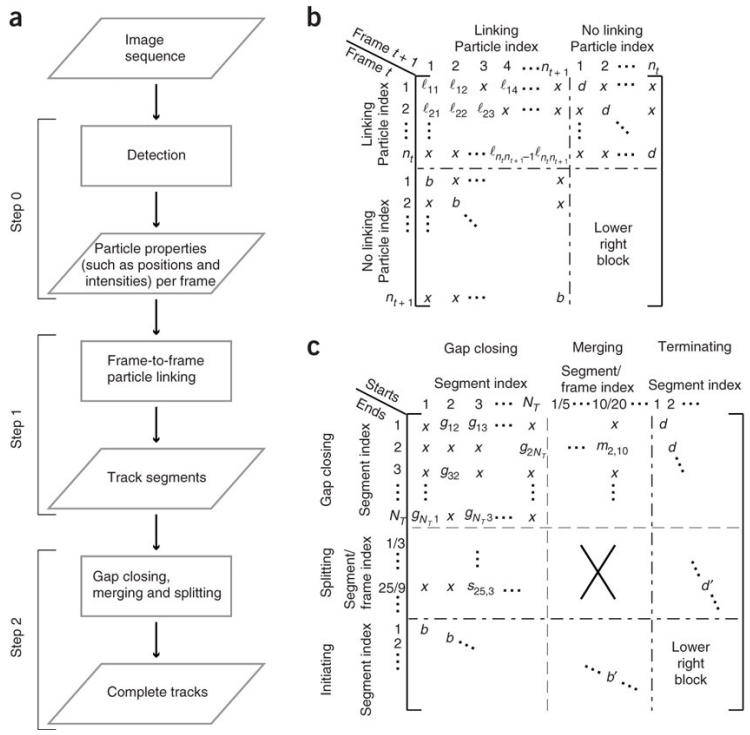


Figure 21: Flowchart of *u-track* tracking algorithm (a), and cost efficient matrices for frame to frame linking (b) and track linking (c). Reprinted from [30].

U-Track[3] is an open source multiple-particle tracking software library written for *Matlab*[4]. The tracking algorithm is explained in Section 4.3. *u-track* can be utilized via a graphical interface with *movieSelectroGUI* in *Matlab* or from the *Matlab* command line with several predefined scripts. It is possible to make a list of movies to analyze and define different parameters for each movie. The detection, tracking and track verification can be started in a queue or one by one, to check the result before continuing. The results are saved in *Matlab* structures for every step in a file. Processing steps, that need to run in between steps of the *u-track* algorithms are implemented in the source code. It is programmed to track dense particle systems with detection failures, splitting and merging events. In addition it considers instantaneous reversal events and has several capabilities of particle detection. Nuclei detection, point source detection and Gaussian Mixture-model fitting are the most relevant for our task. It has also the capability to track in 3 dimensions.

Step 0 in Figure 21a is the starting point. Once the image sequence is pre-processed as described in Section 4.1.1, the nuclei-detection with only the particle

[3]https://www.openmicroscopy.org/site/products/partner/u-track
[4]https://www.mathworks.com/products/matlab.html

diameter in pixels as parameter works great. See Section 4.2 for how the particles are detected. The detected particles are represented by their position and an intensity, ranging from 1 to 255 for the given 8bit gray-scale images.

The next step is to link these detections from frame to frame. The linking is done in two directions, first forwards from the first to the last image, and then the other way around. Trajectories from the forwards search are extended in the backwards search and also new ones will be added. After the backwards search, the algorithms looks one last time forwards, because the search parameters depend on the given trajectory. Beside a search within a radius around the last point, a directed search in swimming direction is used to extend the radius in one direction. The circle is extended in the swimming direction. It is scaled with the swimming speed v of the previous trajectory points and is only extended in a user specified angle α, like Figure 22 illustrates. One variable parameter specifies how many frames the particle needs to move in one direction, before it is classified as directed motion. In addition two different movement phases are distinguished. Slow and fast phases are distinguished with different weighting of the speed. Hence the directed search can be reduced. So when only a Brownian motion is expected in a hibernation state of the bacteria, a normal circular search around the last detection point is sufficient enough. The transition time between the slow and fast phase is specified by the number of expected frames. This search expansion feature works greatly with successive forwards, backwards and forwards linking. It allows the algorithm to find candidate 3 in Figure 22 within the extended search radius. In addition, a backwards search will then allow candidate 2 as a possibility. Every particle has an unique particle index, which allows faster access and improves memory efficiency, because copies are unnecessary. Linking is accomplished with a cost matrix like Figure 21b. l_{ij} is the cost to linking particle i in frame t with particle j in frame $t+1$. If the linking exceeds the cutoff, which is calculated from the search parameters, it is marked as x. d is the cost to link the particle in frame t to nothing in frame $t+1$, whereas b is the cost to link a particle in frame $t+1$ to nothing in frame t. The cutoff here is the maximum number of successive frames for the search, specified as variable parameter. The lower right block in Figure 21b and 21c is just included to satisfy the topology of the algorithm.

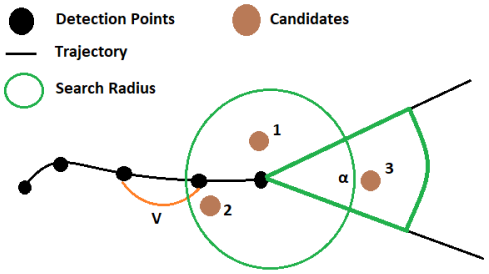


Figure 22: Illustration of the search radius for frame to frame linking

Step 2 in Figure 21 is to link the tracks from step 1 in an optimal way. Again,

indices are used for the linking and a cost matrix like in Figure 21c. Linking tracks is done with gap closing between start and end points of the segments, merging and splitting between different segments at any point and terminating the track if no more segments can be connected. But the merging and splitting is optional in the algorithm, as not all systems tend to do it. For a quasi 2D confinement, merging and splitting the tracks correspond to events when bacteria pass each other. Beside that, they can just collide and repel each other. But bacteria passing by each other can also be tracked with the gap closing. The trajectory is then connected over the crossed bacteria with a gap. Closing a gap between the start of track segment J and end of track segment I has the cost of g_{IJ}. Merging the end of track segment I with a middle point of J has the cost of m_{IJ} and splitting the start of track segment I from a middle point of track segment J has the cost s_{IJ}. The X in the middle of the cost matrix indicates that linking between new track segments from merging or splitting is not allowed. Again, x, b and d are like in the cost matrix in Figure 21b. One last parameter is then a minimal trajectory length for connected track segments.

4.3.2 *U-track* parameters

The first step of the tracking is the bacteria detection. After pre-processing the images, bacteria with equally distributed intensities are detected with the *Nuclei Detection* algorithm in *Matlab*. The radius for the nuclei is chosen as small as 2 pixels or 0.323 μm. Detection rate is already discussed in Section 4.1.1. Tracking needs a lot more parameters. Most of them are mentioned in the previous Section 4.3.1. The chosen parameters and their tested range are shown in table 2. Not all combinations and ranges are tested together. In parentheses are the most frequently employed parameters. Before the parentheses is the tested range for the respective parameter. The success of one parameter combination is a laborious process that was checked by eye and is therefore difficult to illustrate, as it would require a few hundred images to be presented. Using an artificial data set for parameter verification is not possible because every data set needs some adjustments. So for example, parameters like minimal trajectory length or gap closing range depend on the bacteria density in general. And so do the different search radii, but they also depend on the motility of the bacteria. Even the image quality can influence the tracking parameters if the detection of bacteria is fluctuating in some parts of the video.

Parameter	Value in pixel or frames
Minimum segment length	3-8 (5)
Merge	yes/no (no)
Split	yes/no (no)
Split/Merge ratio of intensity minimum	0.3-0.8 (-)
Split/Merge ratio of intensity maximum	1-3 (-)
Split/Merge Search radius lower bound	10 (-)
Directed	yes/no (yes)
Reversal	yes/no (yes)
Multiplier BS	1-3 (3)
min BS	1
max BS	10-20 (15)
BS scaling in fast phase	0.01-1 (0.5)
BS scaling in slow phase	0.01
BS Transition length slow $\leftrightarrow$ fast	2-15 (4)
Use neighbour to expand	yes
Search neighbour in frames	5-30 (15)
Gap closing	yes
Gap penalty	1.5
Max gap to close [Frames]	8-40 (20)
Min segment length for directed	3-15 (5)
Angle for linear search expansion	20-60 (60)
Multiplier for linear search	1-3 (3)
GC Scaling in fast expansion	0.5-1 (0.5)
GC Scaling in slow expansion	0.01
GC Transition length slow $\leftrightarrow$ fast	2-20 (5)

Table 2: *u-track* parameters for tracking for brownian search (BS) and gap closing (GC). The right column shows the tested range; in parenthesis we indicate the best start value.

4.4 Remove sessile bacteria a priori

Before the *u-track* tracking algorithm starts to link detected bacteria between different frames, the data are purged of sessile bacteria and some static dirt. Unfortunately *u-track* has no such an option. If the sessile bacteria are not removed, the search radius in time and space has to be lowered to reduce false linking between motile and sessile parts. Varying light intensity in one recorded video can lead to fluctuations in the detection of one particle. For example, every third frame the particle is not detected, but it is actually there all the time. This can lead to connections between sessile bacteria in the gap closing part of the *u-track* algorithm. If two neighboring bacteria are sessile and the detection of one or both is fluctuating, it could result in a spontaneous detection between those two points. The same thing can happen for a moving bacterium near a sessile bacterium. Removing sessile bacteria in advance has a lot of benefits. In any case it will significantly speed up the tracking process, because the decision if something is staying at the same position does not require as much effort as the decision if something was moving.

The sessile bacteria are saved in a matrix with the dimensions of the video, called *static-sessile*. For every pixel of the images a sessile bacterium can be saved. In addition to the average brightness and standard deviation of the brightness of the bacteria, the standard deviation of their coordinates is saved at the average position of the detected bacteria to *static-sessile*. Actually it is just the average position of the first position points, to save computational resources. There is no need to have the exact average position over the whole video, which would be within the standard deviation of the average position of the first position points anyway. For only some time sessile bacteria or static dirt is saved in a different matrix. For example, if in the first third of the video a sessile bacterium is detected, it is then totally missing in the second third, and is again detected in the last third, it cannot be saved in the same matrix for two reasons. The first is that it is probably not sessile for all of the video and therefore should be treated differently. The second reason is that we cannot simply save two bacteria in one matrix object at the same position, as it is already occupied and a different amount of matrix entries is needed to save all information about this one bacterium, depending on the detection intervals in the video. Hence, intermittent sessile bacteria are saved in a *struct* object in *MATLAB*, called *semi-sessile*, because a *struct* can save varying amount of information for every entry.

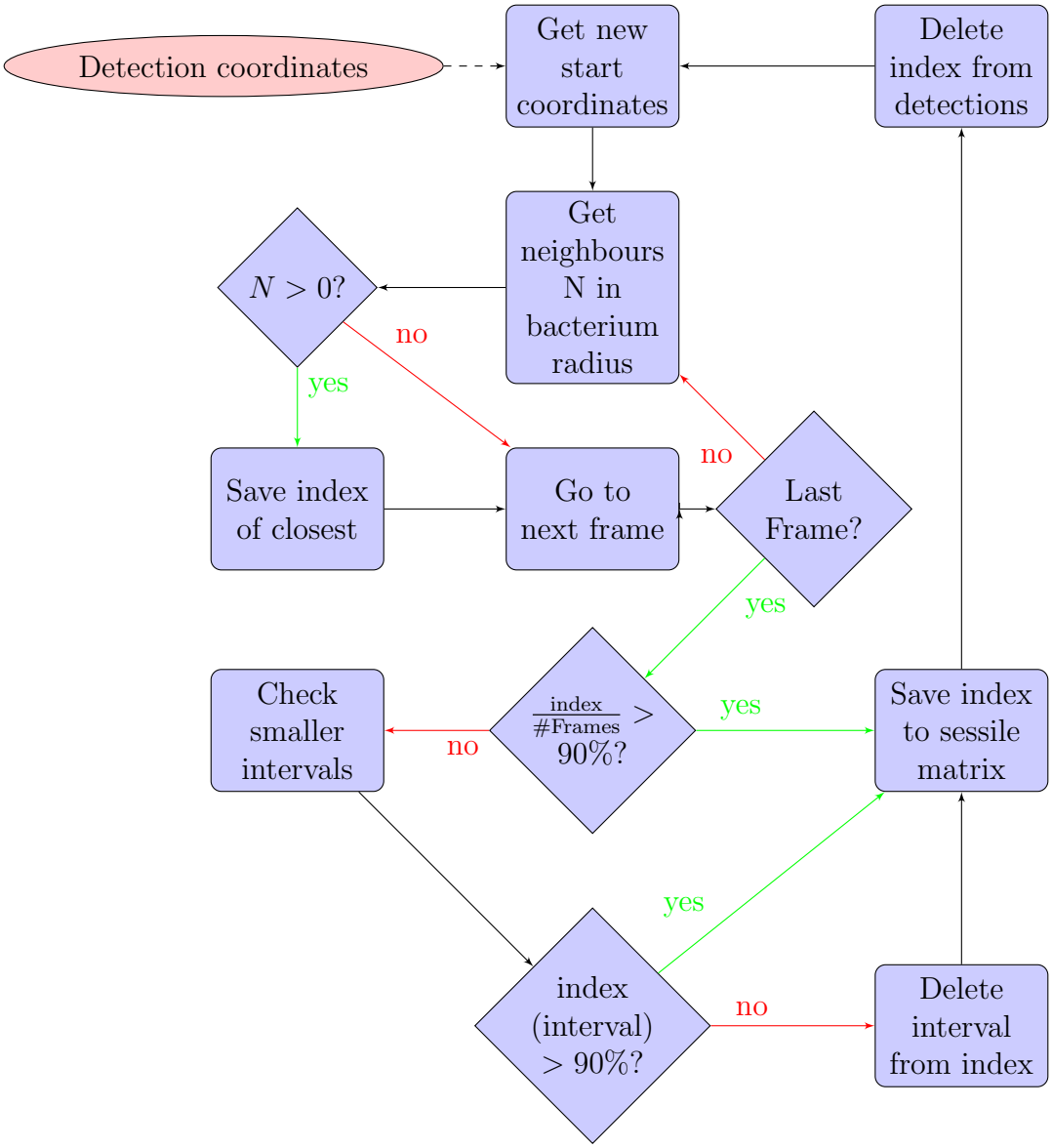


Figure 23: Flowchart of the algorithm to identify sessile bacteria and static noise.

4.4.1 Definition of a sessile bacteria a priori

U-track's Nuclei Detection (described in Section 4.2) saves the position and brightness of the detected bacteria for every frame. In Figure 23 we show a flowchart of the algorithm to identify sessile bacteria and static noise which is described in this chapter. The algorithm starts with the first coordinates in the first frame. The next frame is checked in a radius of one bacterial size around the first coordinates. If there are multiple detection in this radius, only the closest detection is used. The search continues with the following frames until all frames are checked for the specific radius around the first coordinates. Two different methods are used to declare a detection as a sessile bacterium over the whole video at the same position. If there are detections for more than 90% of the video, it is declared as a sessile bacteria

over the whole video and all detections will be deleted from the database. If this is not the case, different time intervals of the video will be checked. For that the video is split into intervals of $5-10\%$ duration. If the detection rate in one of these intervals is above 90%, the interval is marked as a sessile bacterium for this location and all detections in this interval are removed from the database.

Typically a video is either 5 seconds or 50 seconds long, where a 50 seconds video often is split down to something in between 10s or 20s parts. Therefore the intervals of $5-10\%$ become something from $0.25-2$s where 90% is $0.225-1.8$s. If within this interval one bacterium moves out of the search radius, it corresponds to a minimal speed of bacterial size$0.225\text{s} = 12.\overline{4}\mu\text{m/s}$ which is removed. For that reason the interval for very short videos is set to 10% of the video. The minimal removed velocity becomes then bacterial size$/0.225\text{s} = 6.\overline{2}\mu\text{m/s}$. Also a moving bacteria can be sessile for a short period of time in a natural way. *Vibrio alginolyticus* mentioned in Section 3.3 can stop for about 0.1 s when switching from backwards to forwards movement. Therefore the allowed sessile window needs to be at least of this size. With a 10% interval and 90% requirement the minimal removed time window is of 0.45 s.

If 19 of 20 possible intervals are occupied with a sessile bacteria, it is declared as sessile over the whole video and will be saved to *static-sessile*. Otherwise the coordinates and appearance in the video will be saved in the *semi-sessile* object. To reduce computational costs, only the detections of a few frames in each interval are used as starting coordinates. If there really is a sessile bacterium, it should be detected within a few frames, even if it is fluctuating. Checking only 10% of each interval is enough to say that the remaining interval cannot be sessile for 90% with a certain reliability. But there will still be some detections left which are not moving bacteria. In most cases they are false detections which are not removable at this stage and shorter than the sessile window.

4.4.2 Improving trajectories by using sessile bacteria

The *static-sessile* and *semi-sessile* matrix are used to improve the trajectories. If a bacterium moves very close to another bacterium or even overlaps with another bacterium, only one bacterium will be detected as discussed in Section 4.2 and shown in Figure 12a. This detection can be too far off the sessile bacterium's position and hence it is not removed when the sessile matrix is created. It is most likely to be in the center of both bacteria. If the distance from the detected center between those bacteria is smaller than 2 bacterial sizes, it is considered the starting point of a drive-by. All subsequent points will be classified as a drive-by until the distance between both detected centers is greater than 2 bacterial sizes. Bacteria detection

tests from Section 4.2 show that 2 bacterial sizes should be enough to detect their centers. Even if it would be one big white object, it will still be recognized as two detections in most cases as shown in Figure 12a.

Let's consider one concrete example, shown in Figure 24. In the original detected trajectory, the bacterium is considered sessile for a brief moment at the red star in the top image of Figure 24. It is the only detection point within a few frames. After detecting the sessile point, the bacterium continues with a much greater velocity compared to the others. The size of the velocity can be seen by the distance between two consecutive stars in Figure 24. The improved trajectory in the bottom of Figure 24 shows velocities of the same sizes, which can also be observed in a video of the trajectory. As a result the velocities and direction changes are smoother. The downside is that we will lose the information if the bacterium did really stop at the other bacterium. But there is no known biological reason for *Shewanella oneidensis* MR-1 to do so. Observation shows that this method is the right way in almost all of the cases.

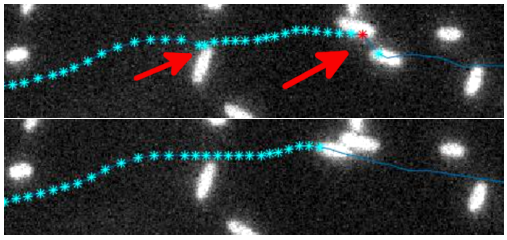


Figure 24: Trajectories of a bacterium moving next to sessile bacteria. Blue stars are detection points. Top: original trajectory, bottom: improved trajectory.

Another advantage comes from the sessile matrix. The majority of the detected sessile bacteria will be actual bacteria and not noise. Just very few are detection errors. Therefore an average brightness I_{avg} and the standard deviation I_{std} of the bacteria can be evaluated, before analyzing the motile bacteria. This is used to remove detection errors in advance. If a trajectory is shorter than 2 bacterial sizes from start to end point and its average brightness is lower than $I_{avg} \cdot 0.5$ or lower than $I_{avg} - 10 \cdot I_{std}$, it is removed from the data set. The low brightness detections are artifacts from the background noise, in most cases. After the image preparation described in Section 4.1, the average brightness can be around $250/255 \pm 4$. So an interval down to $I_{avg} - 10 \cdot I_{std} = 210$ is still very high compared to $I_{avg} \cdot 0.5 = 125$ and visible by eye. Also larger I_{std} can occur.

5 Processing Trajectories

5.1 Cleaning trajectories

U-track has an algorithm to verify if trajectories are real or not. Unfortunately the number of correct trajectories does not go up significantly in a visual check with the given parameter range. Therefore another algorithm, more suitable for these bacteria, is needed to verify the quality of every trajectory. At first the number of interpolated positions is checked. If he number of interpolated positions is over 80%, the trajectory is removed completely. Most importantly a significant amount of sessile start and end points along with interpolated start and end points are removed for moving bacteria. Because of the bacteria density and detection errors along sessile bacteria, which are not catched by the *sessile matrix algorithm* of Section 4.4 and for reasons connected to some motile parts, the tracking algorithm tends to start and end with a sessile point. In addition the bacteria tend to run in and out of the focal plane when colliding with another bacteria. So the trajectories often start or end with a sessile point.

It is necessary to remove the start and end points in a *while loop* until neither a sessile or interpolated start or end point is found. If one of them is removed, the other might appear. For example, an interpolation between several sessile points with an interpolation of significant movement. The first step is to remove trajectories which are shorter than a certain *loop-back* window. This window will be used in following algorithms to verify decisions within the *loop-back* window. In general it is very short, even shorter than the minimum segment length of the tracking algorithm and has therefore no effect at first. By removing points of a trajectory it can become shorter than the minimal length and is then removed completely from the data set. After that the start and end is checked for interpolated positions. If one point within a typical *loop-back* window of 5 frames is interpolated, everything before is removed. The same goes for a sessile start or end. The sum over the position deviation $\Sigma_{xy} = x_{\text{dev}} + y_{\text{dev}}$ is calculated in both dimensions separately. The sum will stay around zero, if it is fluctuating at the same position. If Σ_{xy} does not grow over $2\cdot$ *precision Error* within the *loop-back* window, it is classified as sessile. Sessile start and end points are then removed until Σ_{xy} gets big enough to be classified as a moving part. When neither the interpolation or sessile check finds something in an iteration, the *while loop* ends. The total number of interpolations points is checked again and the whole trajectory is removed when this number is greater than 80%.

After that the distance between start and end points are checked. Furthermore the distance to the next sessile point is also checked. Only sessile bacteria from the *sessile matrix* of Section 4.4 are used. If trajectories happen to start and end near

another sessile point and their distance from start to end point is smaller than 2 bacterial sizes, the whole trajectory is removed.

5.2 Detection Error vs. Movement

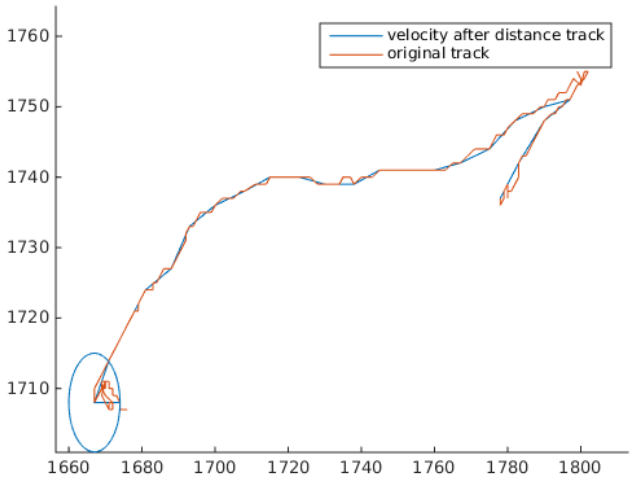


Figure 25: Example of tracking using the velocity after distance‘ method.

The velocity and position vectors are analyzed, to clearly distinguish between real movement and movement due to detection issues. The average measured error for the *u-track* detection algorithm is ~ 2.2 pixel at 20x2x magnification or $\sim 0.35\mu$m as discussed in Section 4.2. Setting it up to ~ 3 pixel or $\sim 0.5\mu$m fits better to the distribution of Figure 20a and gives a buffer for bacterial size changes due to the heavy image processing.

Therefore the bacteria will stay at the same position until it has passed a distance $\Delta x \simeq 0.35\mu$m and 0 velocities are assigned else. We call this vector *velocity after distance* (VAD). The true velocity is the VAD vector divided by the time the bacteria needed to move over this distance. Because of the size of the detection error, it is unclear if the bacteria was moving slow, if the movement emerges from detection errors or just stopped and moved at a fast pace again. The minimal allowed distance Δx, to assign a new position, can be changed easily in the algorithm. Figure 25 is an overlay of an original track and the new track with VAD vector. The original trajectory is fluctuating around the VAD trajectory. The circle at the bottom left shows a sessile part and is almost of the size of one bacterium. The original track is moving a lot within this circle, while he VAD track is not.

Beside VAD there are the simple moving average, the exponential weighted moving average, and the Kalman filter [33] as possible velocity correction method. One

version of a Kalman filter is used for example in GPS tracking for navigation applications. Some are good for moving parts, even better than the VAD, but have issues with no movement or very slow movement. There is still an error according

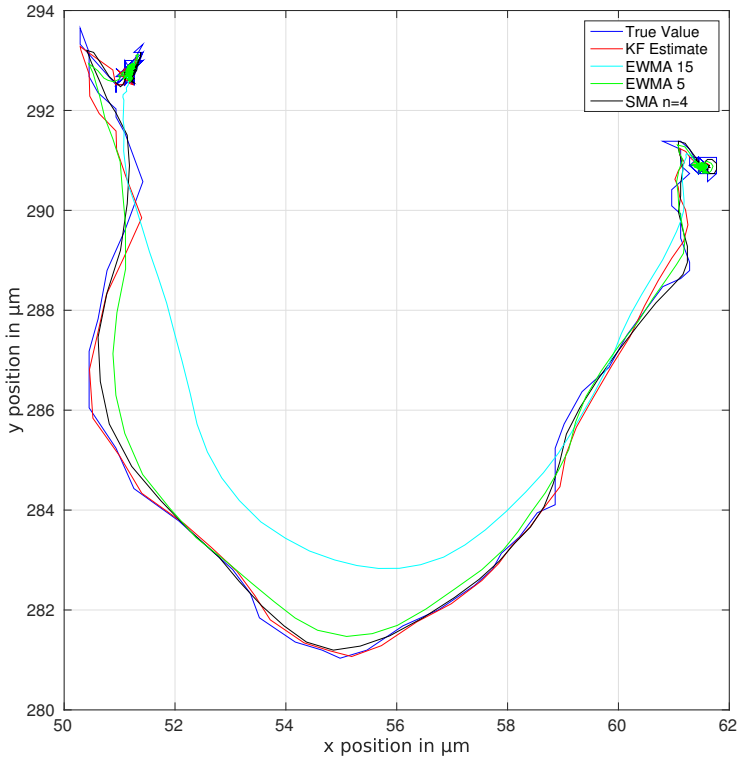


Figure 26: Comparison of moving averages and Kalman filter

to the size of detection error. A simple moving average (SMA) is just the average over the last n data points.

$$v_{sma}(t) = \frac{1}{n}\Sigma_i^{n-1} v(t-i)$$

Depending on the size of n the trajectory may get sluggish. To take newer data points more into account, they are weighted. To weigh them even more, they can be exponentially weighted with a recursive method, which is called an exponential weighted moving average (EWMA).

$$v_{ewma}(t) = \alpha \cdot v(t) + (1-\alpha) \cdot v_{ewma}(t-1), \quad 0 \leq \alpha \leq 1$$

Figure 26 shows an example trajectory with the different filters. An average over a smaller interval gives better results compared to the true value. Taking too many old positions into account lead to a significant offset in non-straight movement. With the right parameters the position can be predicted between frames even while moving in a curve.

5.3 Defining motile and sessile trajectories

After all previous classifications are done, the trajectories are classified as sessile or motile. Dead bacteria and dirt on the glass or camera should not move and have been already removed. The only thing left are bacteria which move on the same spot in a recognizable way. Some are already caught with the algorithm in Section 5.1. One problem is that some trajectories are short and do not cover much distance. Hence they can be easily classified as sessile. To avoid false classification, a number of parameters is used for a decision.

The border between sessile and motile is set with 3 parameters. The first is the average velocity of all moving parts, according to the VAD of Section 5.2. It needs to be bigger than the detection error per frame. The second is the distance the bacteria have passed in total, notwithstanding the direction of the bacteria. A bacterium with a twitching movement of $1\,\mu$m per frame would have a distance of $50\,\mu$m after 50 frames, but such a bacterium will be removed with one of the previous algorithms anyway. The last parameter is the standard deviation std_x and std_y of all positions. It needs to be bigger than the detection error. The sum over all criteria $\Sigma std_x + \Sigma std_y + \overline{VAD} + distance_{\text{total}}$ is called *motility*. The sum needs to overcome a threshold of 50. This combination determines motile bacteria in the following way. A trajectory is motile if all 3 criteria are fulfilled.

5.4 Defining trajectory segments

First Frame	Last Frame	Angle	Distance	Agility	R
1	35	0	52	2	0
36	55	78	31	2	0
56	61	0	2	0	0
62	110	185	27	1	1
111	130	-98	38	2	0

Table 3: Trajectory segments example

The goal is to distinguish between different behavior of one bacterium by defining different trajectory segments. There are more complicated states than only motile and sessile which needs to be addressed. One very important and common behavior is the bacteria being stuck to the glass surface while trying to move. In addition to the biological reason for a direction change, regarding the run-reverse-flick strategy described in Section 3.3, *Shewanella* changes its direction due to other effects, for example hydrodynamic interactions. But the target is to identify only the biological reason for direction changes. A matrix with parameters for every trajectory part will

be created after this analysis. The entries follow the scheme of Table 3. At first the VAD and true velocity vectors as described in Section 5.2 are calculated. Then the angle between consecutive velocities is calculated to detect direction changes. If the bacteria follow the same direction the angle is 0°, −90° if it moves to the left, 90° if it moves to the right and goes to ±180° for a reverse of direction. Figure 25 already showed fluctuations while moving on a small time scale, whereas fluctuations also happen on a bigger time scale. But these fluctuations are actual movement. The direction is changed through collisions, hydrodynamics and possibly other effects. The next step is to correct the trajectory as descried in Section 4.4, before the angles are usable. A moving average with proper parameters could be one way to identify a biological reason for a direction change. However we used a different approach, as finding the right parameters seems not that simple.

Starting from the first detected point of a trajectory, the sum over the angles between consecutive VAD vectors is calculated. A direction change to the left results in a negative angle where a direction change to the right results in a positive angle. If the bacterium is not moving, the angle is set to 0. Therefore left and right turns neglect each other in an approximate straight movement. Calculating the sum has the advantage that a sudden direction change is immediately detected. For example, a standard deviation or any kind of mean over time would change sluggishly. So if the sum is greater than 40° the next 0.05 s or 5 data points at 100 frames/s are checked if the sum stays above 40°. If this is fulfilled a new trajectory segment starts and the sum over consecutive angles starts at 0 again. After all trajectory points are checked and put into parts, the agility of every part is calculated. If the passed distance in one part is greater than one bacterial size and the mean velocity is greater than the precision error, the part is classified as fast. If the average velocity is lower than the detection error, the part is classified as slow. Otherwise as sessile. So only parts with high speed and low distance will be classified as sessile here. Most of the sessile parts should be just a stopping point through the VAD method, Section 5.2.

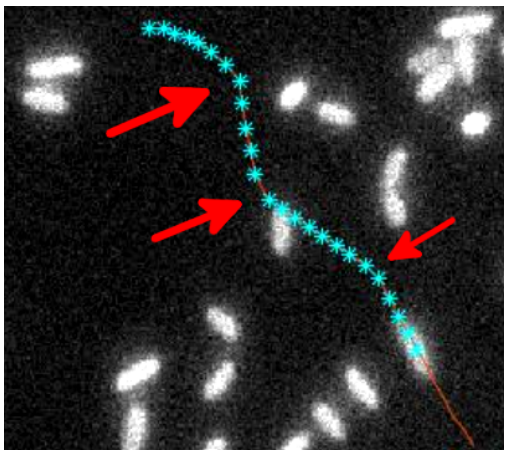


Figure 27: Example of winding movement around sessile bacteria

In the next step, the distance and angle between start and end point of consecutive trajectory segments are analyzed to identify sessile and winding straight

movement. Also the angle between start and end point can lay below 40°, because the sum was greater only for a brief moment. It can be caused by a movement around a sessile bacterium as shown in Figure 27 for example. The trajectory part angle in column 3 of table 3 is always the angle between start to end point of one trajectory part to the start and end point of the previous trajectory part. So the first trajectory part cannot have an angle. First the sessile parts are identified. After analyzing the angle, the distance the bacteria have passed is analyzed. If the distance of two subsequent trajectory segments is smaller than one bacterial size, the standard deviation of the position in this parts are calculated. If the standard deviation in either X or Y direction is smaller than the detection error, the parts are connected together. Also if the standard deviation of the angles in these two parts is greater than 40°, the trajectories are connected. This helps to identify fast movement in a small area as sessile. The part is classified as sessile by setting the agility in the trajectory segments matrix, like table 3 column 5 shows, to 0. When a sessile trajectory segment follows another sessile part, they will also be connected. However, if two motile parts occur in succession with another motile part, they can be connected for several reasons. At first, the angle of a trajectory part is checked. If it is lower than 40°, and also smaller than the angle of the neighboring trajectory part, it will be connected to this neighbour. The agility parameter for all trajectory segments are corrected according to the new trajectory segments with the same classification.

Afterwards the sessile parts need to be split from the motile parts, because the angle is set to 0 if there is no velocity. Hence they are still connected. The sessile movement is only cut off and connected at the end and start of a trajectory part. If the bacterium stops while moving straight, there is no reason to cut it out of this trajectory part. At the end and start of a trajectory part, a direction change is expected and therefore it is a useful information to know if the bacteria does stop before changing direction. Beside the obvious definition of absolutely no movement, another is added to allow small movement on the spot. Movement with a standard deviation in angles above 20° and standard deviation in position below $2 \cdot$ detection error is allowed. The standard deviations are also calculated in a 0.05 s window. If the criteria are fulfilled, the window is chosen smaller to find the exact end point of the sessile part. If a sessile part follows on a motile part, the same definitions is used to look for the exact start of the sessile part. After cutting off the sessile parts, the entries in the trajectory part matrix are corrected according to the new trajectories.

In the next step sessile parts are distinguished from sessile parts with movement. They are connected in the first place because a sessile part with movement can also

have a lot of parts without movement. One bacterium can stay at the same position within a small radius of its own size and can move clearly above the detection error. A possible explanation is that the bacteria is stuck to the the glass surface with either its body or its flagellum. Another possibility are detection errors, which occur because of a bacteria cluster like in Figure 12a or light fluctuations in the recorded video. If a bacterium is stuck over a long period of time within a small region, it should be recognized beforehand in the sessile matrix. So only short stuck phases and stuck phases within a bigger area remain to identify.

6 Fundamental statistics of Shewanella

The previous chapters have described a method that yields good trajectories for *Shewanella oneidensis* MR-1. Simple statistics of their motion can be obtained with them. At first the conservation of bacterial number during the recorded video

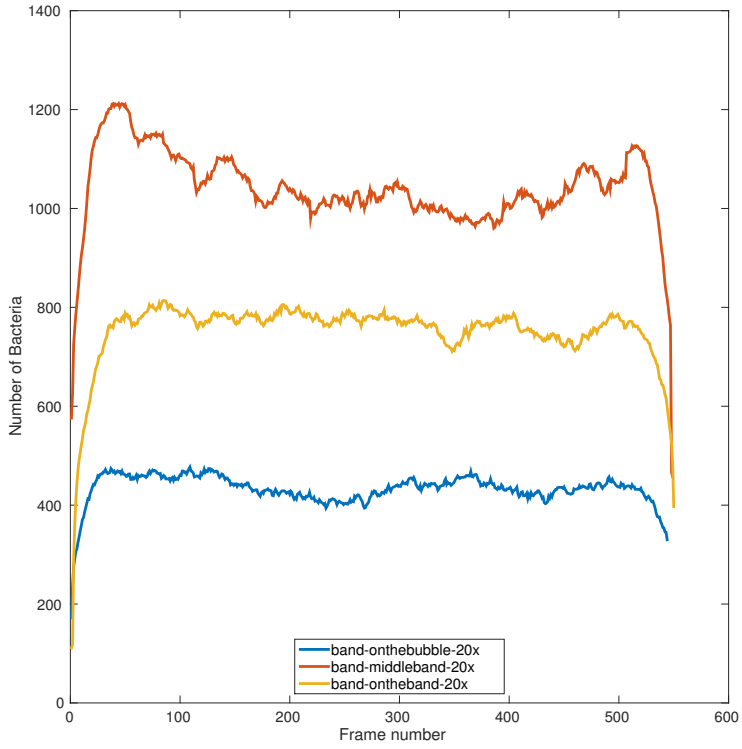


Figure 28: Number of detected motile bacteria over time for all 3 data sets from bubble to band

is checked. Figure 28 shows the number of motile bacteria during one video. This looks similar for all videos, but the total number is of course different form video to video. The number swiftly goes up to a steady amount with fluctuations around a stable value until it drops at the end of the video with the same swiftness. The drops are most likely linked to the tracking algorithm, as it takes time to identify

movement correctly. A stable number of moving bacteria can be expected in a small time interval of 5 seconds. If the number of motile bacteria has a different behavior, a wrong choice of tracking algorithm parameters can be expected. For example, a growing number of motile bacteria is linked to spurious connections to sessile bacteria. Bacteria that disappear are linked to sessile bacteria and therefore stay in the statistics of motile bacteria. But this example is already removed as described in Section 5.1.

The next step is to check the bacteria density. Therefore the interval method described in Section 4.1.4 is used to split the data sets in 13 intervals like Figure 14 illustrates. The density is then the number of bacteria per interval, divided by the respective area. 13 intervals give a ΔR of about 25 μm or 9 bacterial sizes per interval considering the long axis. In addition to the motile bacteria, the sessile bacteria are also analyzed. For the number of sessile bacteria, the sessile matrix entries and sessile classifications afterwards are used. In general they outnumber the motile,

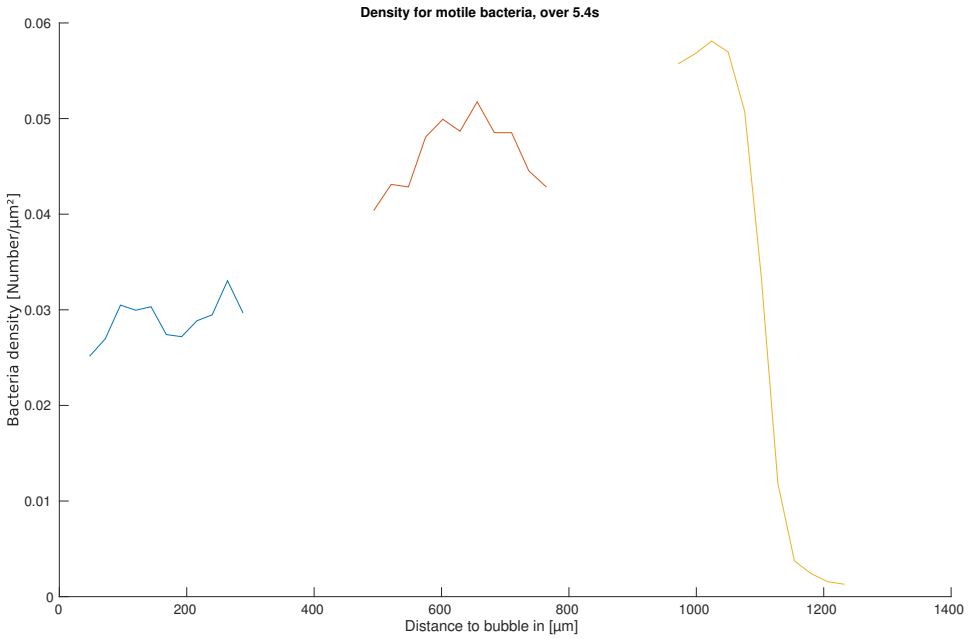


Figure 29: Dependence of the density of motile bacteria with respect to the distance to the bubble.

just like in the temporal analysis before. Motile, sessile and their total sum is shown in Figure A.5. Drops at the first and last point seem to happen in all videos and are probably related to a systematically error. Therefore they are removed from the data set and only 11 of 13 intervals are used. Every color corresponds to one location, with 2 recorded videos each. The gaps between the different colors are a result from the spacing between between the locations as described in Section 4.1.4. No data are available in the gaps. Figure 29 shows the bacteria density only for

motile bacteria, because it is hard to see in Figure A.5. A continuous growth in motile bacteria with a sharp drop at the bacterial band edge is observed at around 1100 μm.

Further statistics are obtained with a steady number of motile bacteria and 2 data sets for every location with respect to the bubble, which are just a few seconds apart. The distance to the bubble for different videos is obtained as described in Section 4.1.4. In Figure 30 three locations are recorded from bubble to the aerotactic band. The first is on the bubble, the second in between and the third on the band. The first property is the velocity distribution of motile bacteria. To reduce the error, the VAD vector described in Section 5.2 is used here. Every velocity of each trajectory is used, which means a trajectory with 50 steps results in 50 entries in Figure 30. If the bacteria move less than the minimal distance allowed (Δx), the distance and needed time is used as velocity. All velocities larger than Δx/frame have detection artifacts due to the high frame rate. These artifacts are clearly visible in Figure 30 with a small $\Delta x = 0.31\,\mu$m. Different Δx are analyzed to obtain good statistics. Velocities

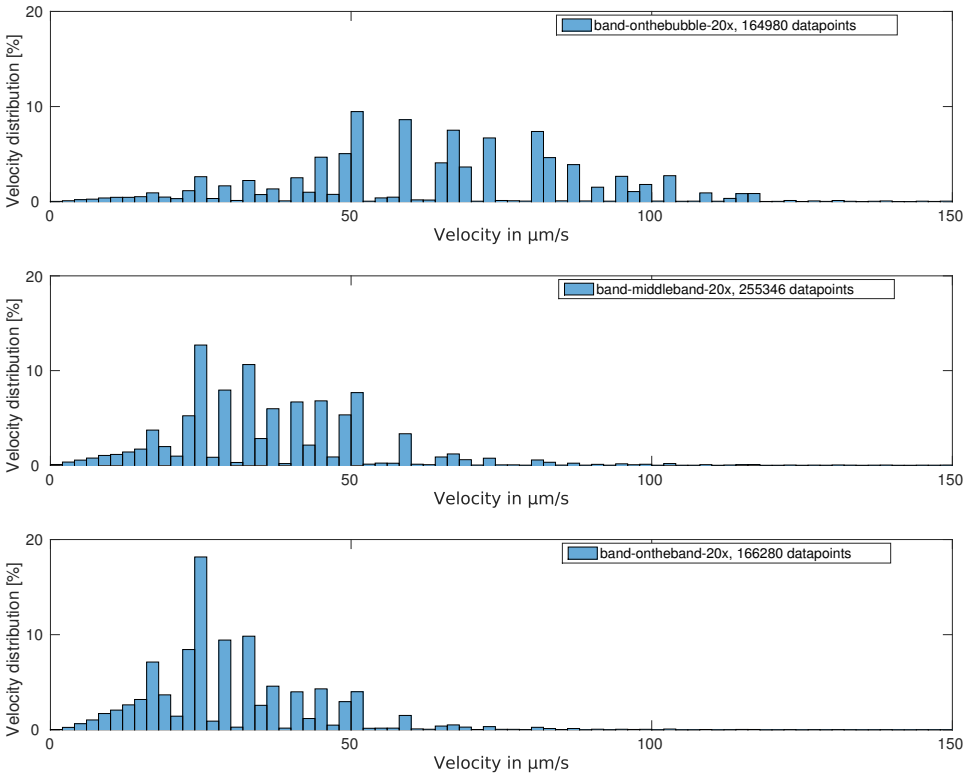


Figure 30: Velocity distributions for velocities with movement above detection error, $\Delta x = 0.31\,\mu$m per frame. Top, middle and bottom panels correspond to locations on the bubble, in between and on the aerotactic band, respectively.

up to 100 μm/s = 6.2 pixel/frame are expected and will therefore show artifacts from the detection. Some velocity peaks correspond to velocities of $n$pixel/frame with n as positive integer. This is very good visible for the velocity distribution with the

minimal time step of one frame, Figure A.11. Therefore the minimal distance for a velocity vector changes from 0.31 μm to 1.4 μm or about half a bacterial size in Figure A.8, to cover the whole range of velocities up to 129 μm/s. An average velocity of about $80, 40$ and 35μm/s with a wide distribution up to single values at 150 μm/s are observed. Since the average trajectory length in time is significant smaller than the video length and the trajectory length is varying in general, the average velocity for every trajectory is shown in Figure A.10. The trajectory average looks similar for different Δx. Sometimes the trajectory of one bacterium is split into several parts because of tracking issues. Therefore a different trajectory does not necessarily imply a different bacteria but in most cases it should. With a bigger $\Delta x = 5.6\,\mu$m, the distribution of all velocities in Figure A.9 converges to the distribution of the trajectory average. It can be assumed that small affects within one trajectory start

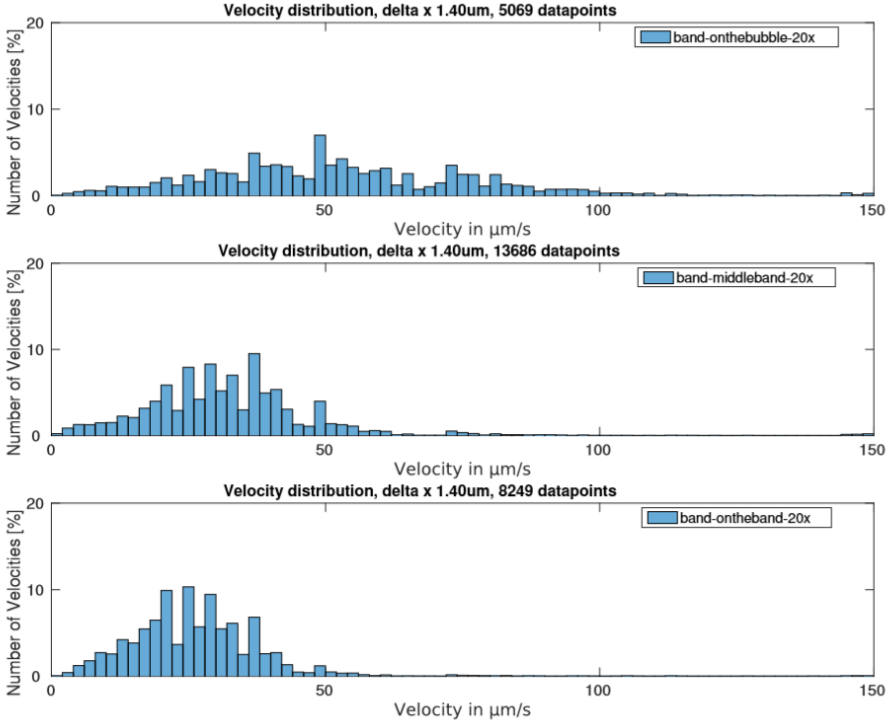


Figure 31: Velocity distributions for velocities with movement above $\Delta x = 1.4\,\mu$m per frame. First velocity below Δx are always deleted. Top, middle and bottom panels correspond to locations on the bubble, in between and on the aerotactic band, respectively.

to disappear above $\Delta x = 5.6\,\mu$m.

Beside an average over time, an average over the distance to the bubble is calculated. All vectors over the whole video within the respective interval are used to calculate the respective average. If a trajectory hits 2 or more intervals, it contributes to all areas with the respective part. Hence 13 data points are collected for

every location with respect to the bubble. Figure A.14 shows the result. The last data point is cutoff in all 3 locations because the last area was significantly smaller and contained few data points. In addition to the average, a standard deviation is calculated and shown as an error bar. The size of the error bar is a result from the wide velocity distribution in general and is stable over the whole recorded distance, where the average velocity goes down with the distance to the bubble. The distribution at a certain distance only grows smaller when the amount of data points goes down as well. This is often the case at the video borders.

Consecutive velocity vectors are connected under a certain angle. The angle was already used in the trajectory part classification in Section 5.4 as an attempt to identify different movement directions. A movement strategy to morphologically similar bacteria, as described in Section 3.3 is assumed to classify trajectory segments. To check preferences in directions only the distribution of consecutive angles without any assumptions is examined. Velocities with $\Delta x = 5.6\,\mu$m are used to calculate the angle, because it works best with the velocity distribution compared to the trajectory average. The absolute angle of all consecutive velocity vectors are shown in Figure A.16. A clear peak for an angle around 0° with an exponential decay over all possible angles is observed. No clear peak follows at this spatio-temporal resolution. Artifacts are again visible for small steps in Figure A.15 and A.17, just like they are visible for velocities with small steps. It shows that the majority of measured movement is straight and therefore distributed around an angle of 0°.

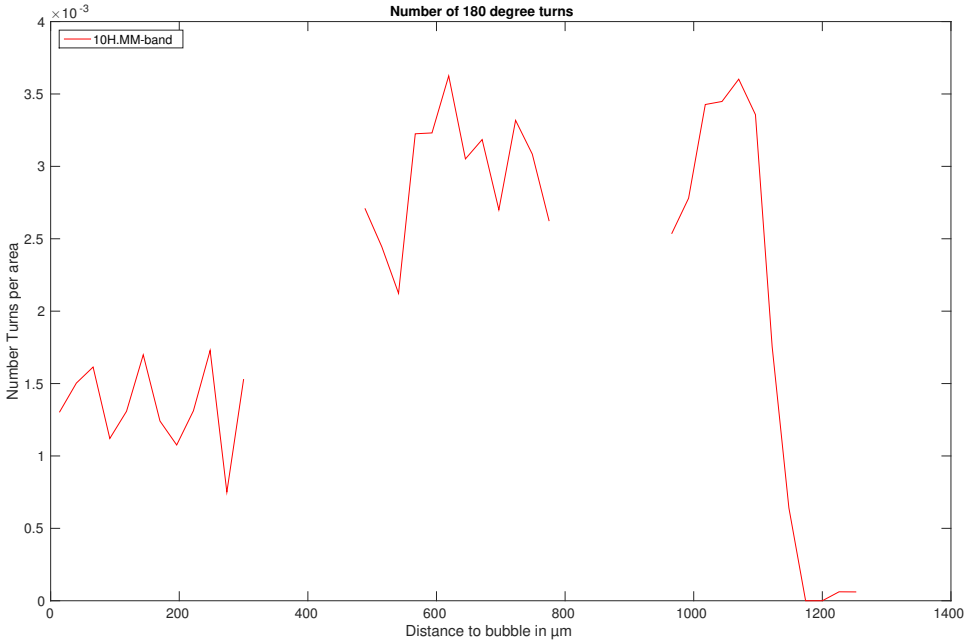


Figure 32: Number of 180 degree turns between consecutive trajectory segments.

In addition to the relative direction change, the direction change relative to the

bubble surface is analyzed. The angle is calculated relative to the bubble surface inwards normal. With the bubble at the bottom of an image, movement straight away or upwards is $\pm 180°$, towards right are negative angles or straight to the right is $-90°$, straight towards the bubble or down $0°$ and towards left are positive angles or straight towards left is $90°$. The probability distribution of relative angle to the bubble inverse normal with $\Delta x = 5.6\,\mu$m is shown in Figure 33. In addition the angles can be distinguished between movement towards the bubble or away from the bubble. If the end point is closer the the bubble than the starting point, its is

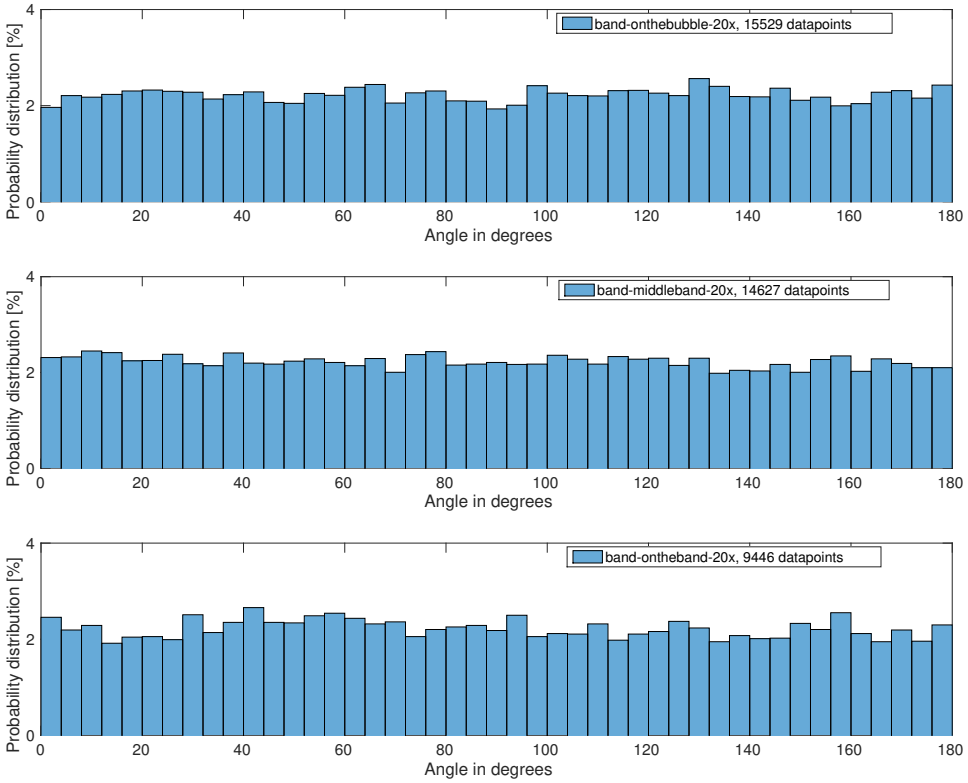


Figure 33: Probability distribution function of the relative angle to the bubble inverse normal with minimal velocity of $\Delta x = 5.6\,\mu$m on the position.

classified as inwards and outwards the other way around. Start and end points from the VAD method are used with $\Delta x = 5.6\,\mu$m. Figure A.20 shows the result for small space steps with the VAD method and Figure A.21 for small time steps. Here the artifacts have a pattern of $n \cdot 45°, n \in \mathbb{N}$ with higher peaks at $0°$ and $\pm 180°$. The peaks in Figure A.21 are a bit too large to fit them in the graph, compared to Figure A.20. Higher peaks at $90°$ indicate movement along the x-axis and $180°$ indicate movement along the y-axis. If we take the small steps into account, the movement is very likely just one pixel. The peaks at $45°$ and $135°$ are then explained with small movement along both axis. The continuous distribution around $n \cdot 45°$ is a result of the bubble shape and the respective inverse normal.

Again, the spacial distribution is also analyzed in Figure A.23 with the standard deviation as error bar to show the continuous wide distribution over space. Therefore we can assume that the distribution of Figure 33 is similar over the the distance from bubble to band. The absolute values are shown in Figure A.23 because inwards and outwards are equally distributed in positive and negative angles which would result in an average angle of around 0° everywhere. But still the values for inwards and outwards are fluctuating in the middle of their possible angle range. Fluctuations of the inwards movement is higher compared to the outwards movement.

Trajectory segments of Section 5.4 can be used to differentiate between a 180° and 90° turn. Reversal and flick events are distinguished although not all of them might be identified correctly. The identified turn events are assigned to one of 13 intervals as before and normalized to the respective area. Figure A.24 shows the number of 90° turns and Figure 32 the number of 180° turns with an allowed range of ±40°. The bacterial band end is again clearly identified with a hard drop to the end of the recorded area at around 1100 μm. But a peak appears in Figure A.24 at the middle of the recorded data set in contrast to the density of motile bacteria. Whereas Figure 32 shows a more or less continuous high number of reversals at the middle until the bacterial band ends.

7 Conclusion and discussion

7.1 Tracking

Overall the tracking result is satisfactory. Putting a comparatively large computational effort in pre-processing the images yield a good detection rate of $\sim 95\%$. Unfortunately the high precision in time and space leads to an inaccuracy in the detection. This can be clearly seen in the precision error distribution of identical, artificially placed bacteria in Figure 20. Due to the high frame rate of 100 frames per second the relative small displacement of 6.2 pixel/frame, corresponding to a speed of 100 μm/s, has an error of 2.2 pixel/frame or 35.5%. Taking into account speeds lower than 100 μm/s leads to an error even above 100% with speeds below 35.5 μm/s. To reduce this roughness some kind of moving average is necessary to obtain useful statistics. The tested moving averages have all benefits on their own, but a space-wise average seems the most reasonable. The next possible, but untested, improvement is a combination of different averages. Coupling a moving average when the bacteria move and the space-wise average when they barely move looks promising and rational. The upside of the high precision in the first place is the significant better tracking result. Assuming for bacterial sizes of about 3 μm on the

long axis, according to Figure 13, moving bacteria will have a partially overlapping image in consecutive frame for speeds up to 300 μm/s, which *Shewanella oneidensis* is very unlikely to reach. This is even true for the small axis with an average just above 1 μm. Due to the overlapping in consecutive frames, it is a lot easier to make the correct connections for the algorithm.

In addition to all the pre-processing steps described in Section 4.1.1, an image subtraction of consecutive frames is a reasonable attempt to identify movement. But due to the high precision in time it is more adequate to use the same frame or some kind of sessile background summation over a number of frames as a subtraction mask. Otherwise a higher position uncertainty resulting from the overlapping bacteria is to be expected. But this attempt removes the information about sessile bacteria and bacteria sticking to the glass, and therefore possible collision candidates and other resulting information. Nevertheless it should improve the trajectories of moving objects overall.

Tracking parameters are very similar for most recorded videos. The tracking works perfectly for movement without obstacles even with missing detection points. False positive detections and trajectory splitting only happen when collisions with obstacles take place. The most important adjustments are the different search radii and gap closing parameters, listed in Table 2. By analyzing a fair amount of videos regarding the tracking parameters, it is likely that the false connection rate scales with the bacterial density. Dense systems yield better results for smaller search parameters in space and time. To connect trajectories which pass over another bacterium it is necessary to search beyond one bacterial size to connect them correct. Therefore, increasing search parameters results in better trajectories until too many bacteria are within the search parameters. A moving bacterium in really dense systems might just have few detection points between sessile bacteria, which *u-track* only seems to identify correctly up to a certain threshold. Unfortunately, the merge and split ability of *u-track* does not lead to better results as one could assume. All tests were made without removing the sessile positions as they are supposed to merge and split with them when passing over each other. A possible chance for better tracking results is a combination of the sessile matrix approach of Section 4.4 in combination with unused detection points. A stitching algorithm for all motile trajectories, unused positions and sessile positions can easily connect bacteria passing over another bacterium by knowing the positions of sessile bacteria. The ensemble of possible trajectories is significantly smaller when doing the stitching with the mentioned steps in advance. With an approximation bacterial size, swimming direction, and speed the next most likely detection position can be guessed and connected if there is a candidate. In addition an improved detection algorithm is

very helpful. At this stage the detection algorithm only determines the position and intensity of a bacterium. The shape and orientation of the oval shape can be a useful attribute to make the right connections. Of course, it can change due to the third dimension perpendicular to the focal plane, but it should happen in a comprehensible way due to the high frame rate and with the knowledge of sessile positions. When a wider range of bacteria intensity is allowed during the image pre-processing, overlapping bacteria can also be identified by their intensity. Along with a new detection algorithm the detection precision-error should be addressed.

7.2 Statistics

The problems identified with the tracking are also visible in the statistics. The number of sessile bacteria outnumber the motile bacteria in all recorded experiments. This supports the thesis to remove the sessile bacteria before tracking the motile, to reduce false connections between them. An artifact of the interval method to determine sessile particles for the matrix in Section 4.4 can be identified. Steady number of particles for the assigned intervals are observed in very dense systems. Figure A.7 shows a system with almost 1 bacteria per $(\mu\text{m})^2$. Because of the small variation from interval to interval, their size seems small enough. But the constant drop over the recorded time suggests space for improvement of the algorithm. The number of motile bacteria goes down at the start and end of every video. Hence the data set should be cutoff to have a stable number for some statistics. This drop is linked to the minimal trajectory length parameter of *u-track*.

The velocity distributions show artifacts due to the high frame rate. Those artifacts disappear with a different definition of the velocity as opposed to the velocity between subsequent frames. The space-wise definition seems the most reasonable to overcome the discussed detection issue. Comparing the distribution of the average velocity over the trajectories, Figure A.10, with the distribution of every velocity during the trajectory reveals a significant higher number in small velocities (see Figure 30). As the average over the trajectory does not show a significant amount of small velocities, it can be assumed that a noticeable amount of trajectories move partly slow. On the other hand the small velocities can be an artifact from the VAD calculation. If the bacteria stop and start to move again, the first velocity will be a lot smaller because the sessile time is used to calculate the velocity. If the bacteria do not really move, and then start to move again, the time for velocity calculation is too large and results in a incorrect velocity. Slow movement is not clearly distinguishable from detection errors as discussed before. The small velocities are below $35.5\,\mu\text{m/s}$, which is the detection error. Reducing the time resolution, for example

taking every second/third frame, will not reduce this error due to the detection problem itself. In fact a more misleading issue appears. The amount of slow velocities grows as Figure A.13 shows. On the one hand, an average over time when a bacterium is not moving results in slow velocities because of the detection error. On the other hand even with a perfect position detection the bacteria naturally stops and starts to move again, which results in slow velocities. Everything below the respective Δx threshold can either be movement or a detection error. Movement below this threshold can only be specified if the bacterium moves for more than Δx over a distance of at least $2 \cdot \Delta x$. If the bacteria cover a detectable distance with low speed, multiple velocities below the $35.5\,\mu$m/s threshold should follow subsequently in a perceptibly amount of trajectories. Figure 31 shows the velocity distribution when we remove the first value of the speed below the threshold. Different Δx are tested to confirm the velocities below the detection error. If the bacteria move back and forwards in a Brownian fashion with a typical amplitude smaller than Δx, this motion will not be visible when the search radius Δx is used. Slow velocities still appear more frequently in Figure 31, when more than one slow velocity follows on another. $\Delta x = 1.4\,\mu$m is more appropriate, because with too large Δx, all velocities will also be a time average. Figure A.8 and 31 show also a higher number in small velocities when subsequent slow velocities are removed. A comparison of Figure A.9 and Figure A.10 shows that the distribution of all velocities within a trajectory approaches the trajectory average for large enough Δx. Figure A.12 and A.10 indicate an identical trajectory average velocity for the space and time wise filtering.

The detection of the directionality of the motion is also subtle. Consecutive angles show no preference for a preferred direction. A rather fast decay is observed in Figure A.16 whereas A.19 shows some preferences. Therefore the majority of movement can be assigned to straight movement with a continuous distribution of direction changes due to obstacles and other non-biological reasons along with the biological direction changes. As already discussed in Section 5.4, direction changes happen due to different reasons. Statistically, identifying one reason is not simple. But for the tracking some assumptions have to be made. If we use small space or time steps, errors in the detection of the angles will appear as peaks in the angle distribution at values $n \cdot 45°, n \in \mathbb{Z}$. In addition the angle of small vectors also results in $n \cdot 45°$ because the small vectors connect nearby pixels of the image. A comparison to another bacterium like *Vibrio alginolyticus*, mentioned in Section 3.3 can be useful. *Vibrio alginolyticus* exhibits a noticeable stop of 0.1 s to 0.3 s in average, due to its run-reverse-flick motion. It is likely due to the switching time for the motor protein between clockwise and counter-clockwise rotation. The trajectory segments approach of Section 5.4 takes this and several other things into account.

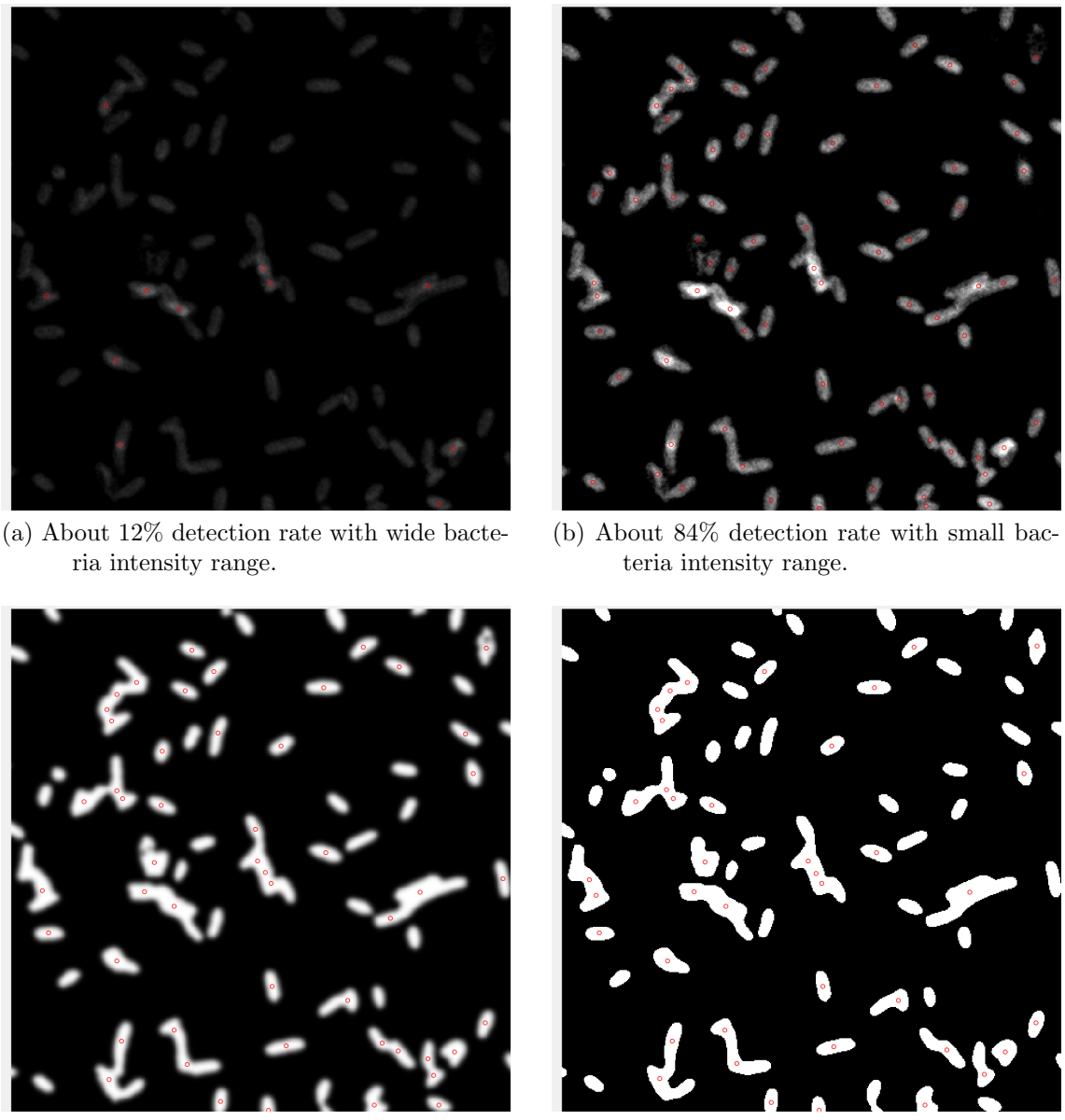
Figure A.19 shows the distribution of consecutive trajectory segments. Again, some preferential angles are detected.

Bacteria movement in relation to their oxygen source is analyzed by measuring their relative movement direction with respect to the bubble surface of the average angle (see Figure A.23). The bacterial band is identified again with a sharp drop at the bacterial band end. But this is most likely an effect due to the drop in bacteria density itself which results in a low number of data points. It can be assumed that the number of data points is too low for statistically robust statements. Higher fluctuations in the inwards movement observed in Figure A.23 could indicate an effect of a rising oxygen gradient towards the air bubble. To confirm this, the effect should be seen throughout the data sets.

In conclusion, we can assess the statistical robustness of the tracking of bacteria movement as good. Problems and artifacts resulting from the high frame rate are purposefully resolved for each statistical analysis. Uncertainty of possible biological origin for the angle distributions remain. Movement is clearly identified and distinguished from non-movement and as a result the edge of the bacterial band. Although the sessile bacteria outnumber the motile by a factor of around 10, the band of motile bacteria can be clearly tracked. Future, possible improvements of the tracking algorithms will improve both the trajectory detection and the statistical analysis. For example, with a smaller error in the detection process, sessile movement is easier identified. For now bacteria stops can only be identified if they are comparable long. Otherwise it is difficult to tell if a short stop is a result from detection issues or a real stop. The detected trajectories can be improved with a stitching algorithm to connect splitted trajectories. This helps to create possible long term statistics and analyze individual movement behavior of different bacteria in a dense system.

8 Appendix

A Additional measurements



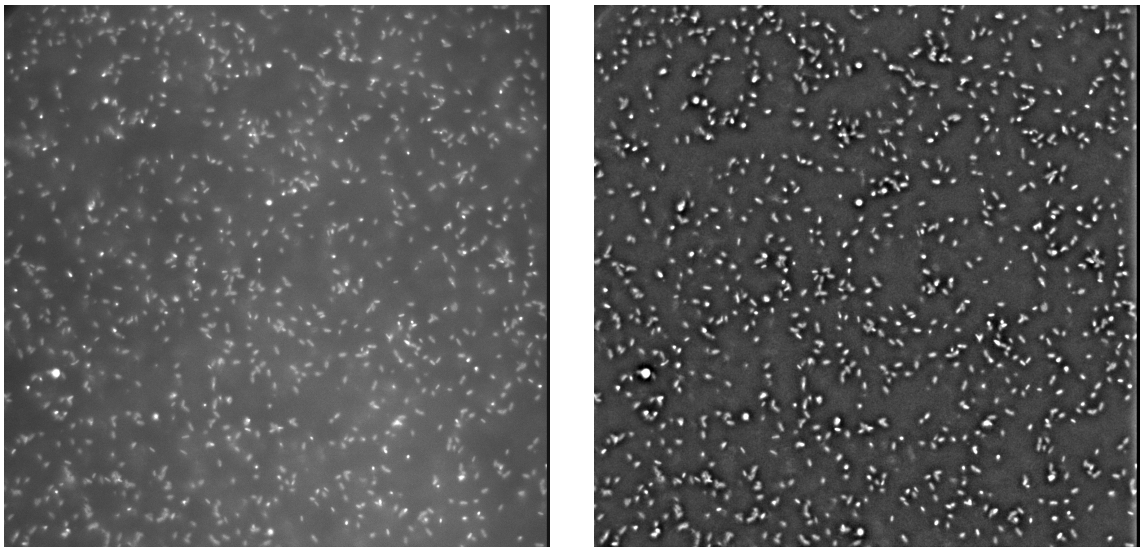
(a) About 12% detection rate with wide bacteria intensity range.

(b) About 84% detection rate with small bacteria intensity range.

(c) About 61% detection rate with GaussianBlur.

(d) About 44% detection rate with a binary image.

Figure A.1: Detected bacteria marked with a red circle after different processing steps. 90 bacteria should be detected here.



(a) Raw image. (b) After band-pass filter.

Figure A.2: Image A.2a shows the image before and A.2b after a bandpass filter of 3-40 pixels. Selected region shows diffuse bacteria, which are more clear after filtering.

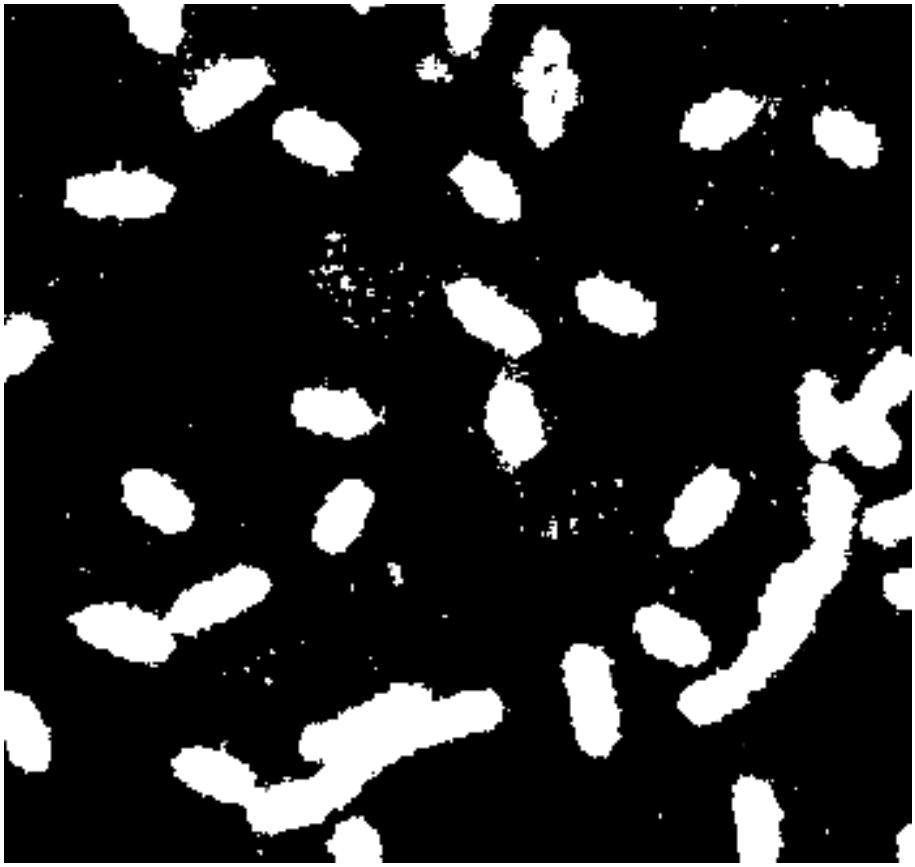
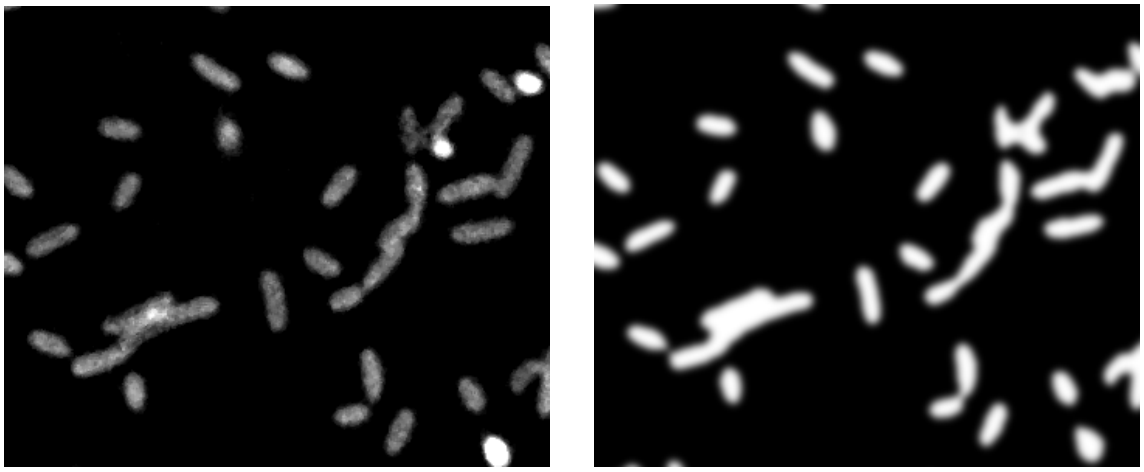


Figure A.3: The small white regions are examples of artifacts after a rolling background subtraction.



(a) Bacteria before Gaussian-Blur (b) Bacteria after Gaussian-Blur

Figure A.4: Comparison of bacteria intensity with (a) and without (a) Gaussian-Blur.

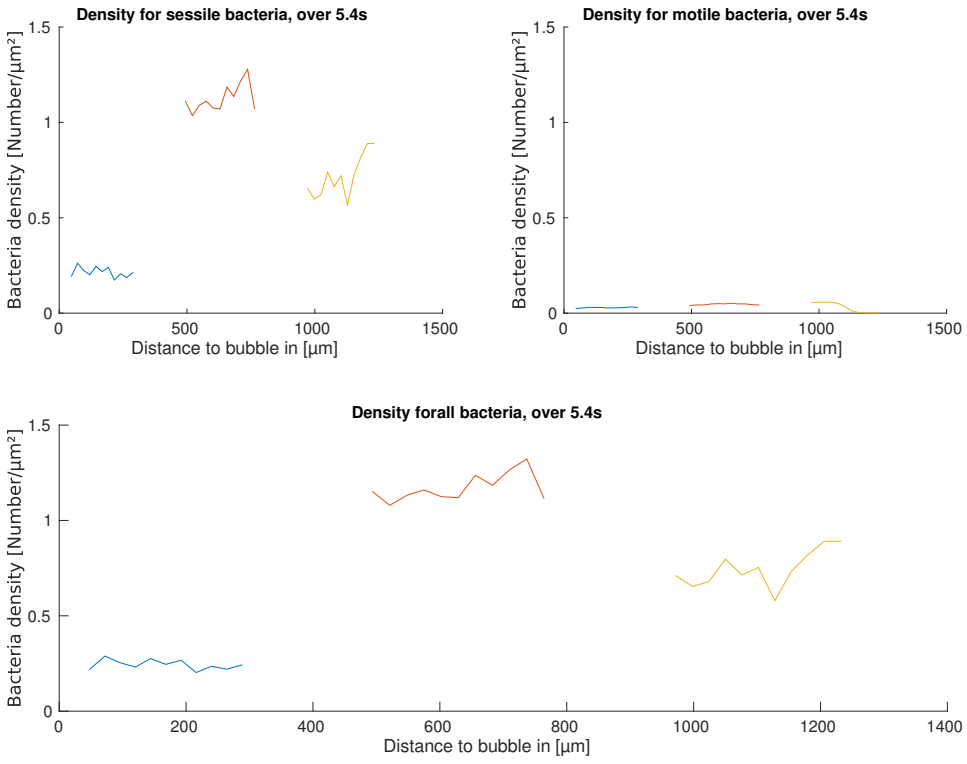


Figure A.5: Density of motile and sessile bacteria.

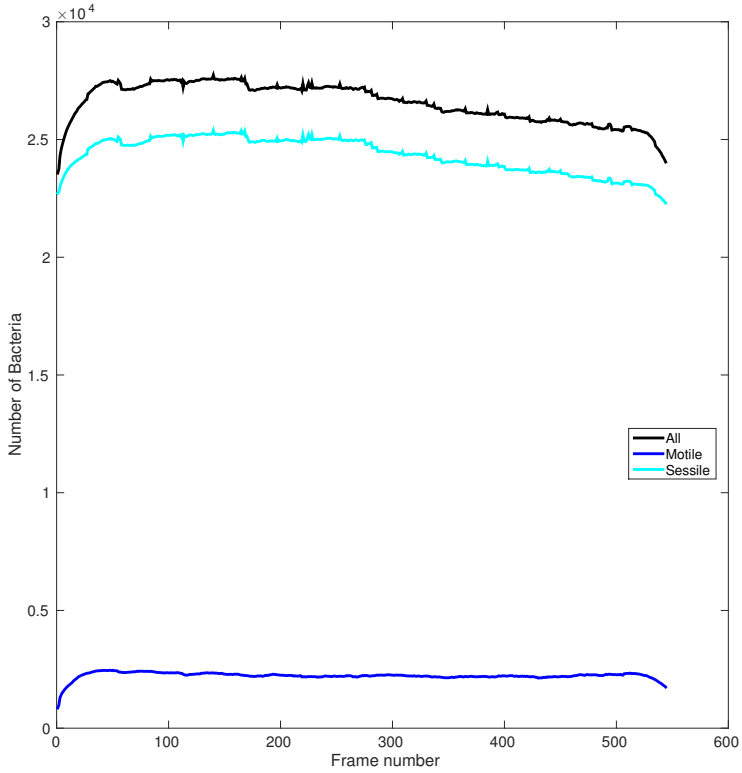


Figure A.6: Number of all detected bacteria over time for all 3 data sets from bubble to band.

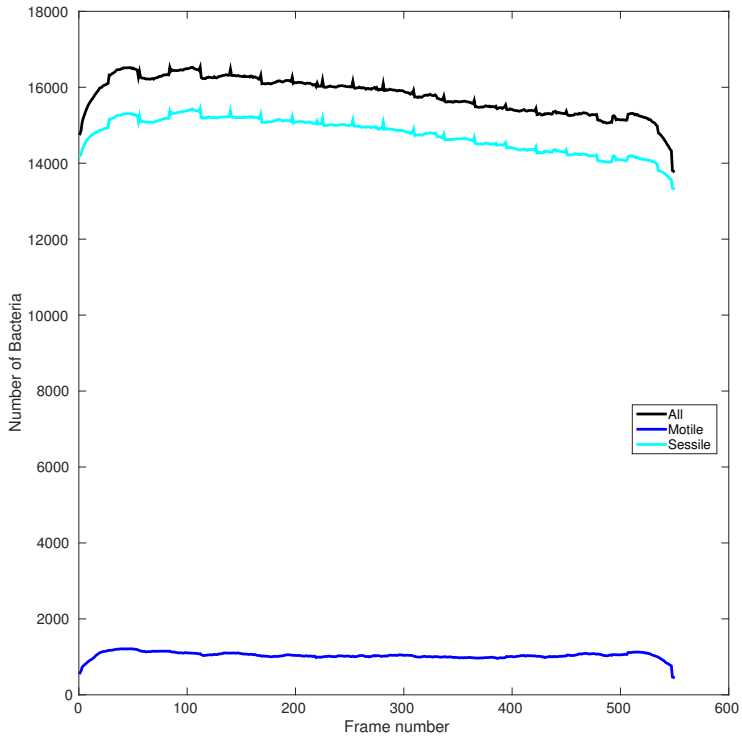


Figure A.7: Number of detected bacteria over time for the data set in the middle. Stairway fits the interval length of the the sessile matrix.

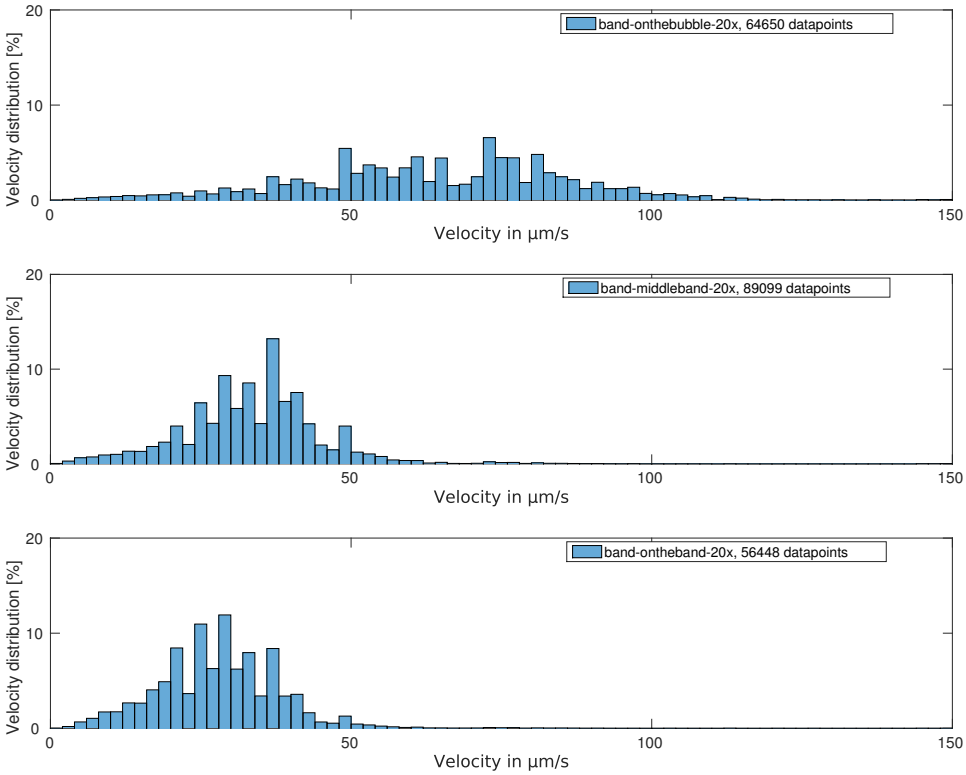


Figure A.8: Velocity distributions for velocities with movement above $\Delta x = 1.4\,\mu$m per frame.

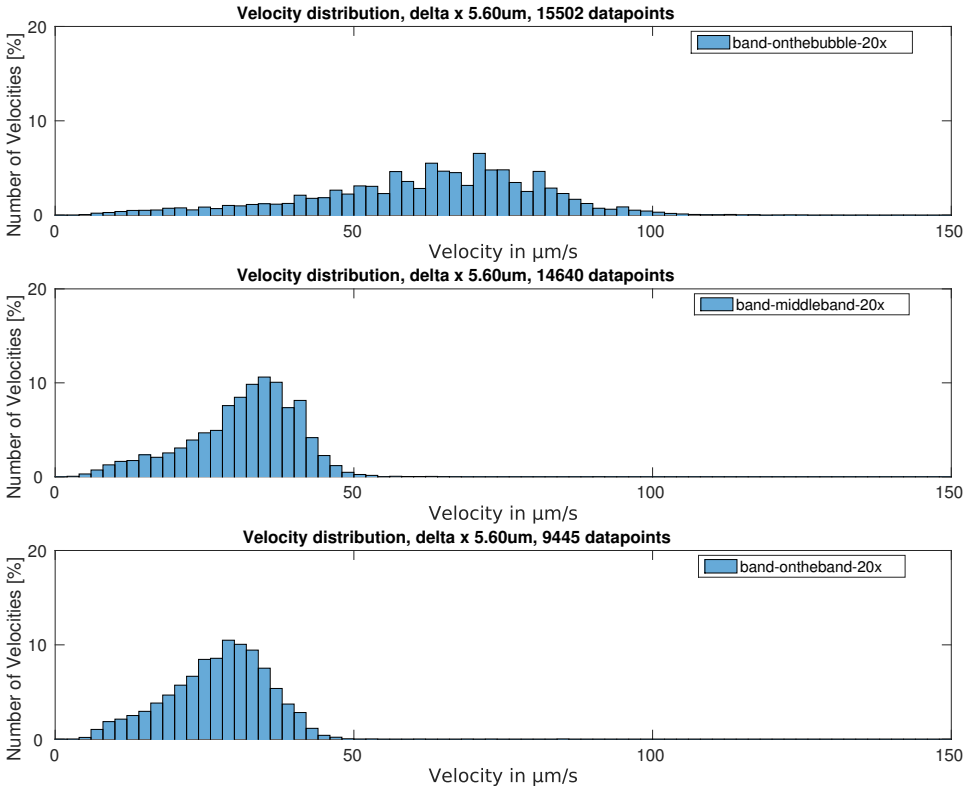


Figure A.9: Velocity distributions for velocities with movement above $\Delta x = 5.6\,\mu$m per frame.

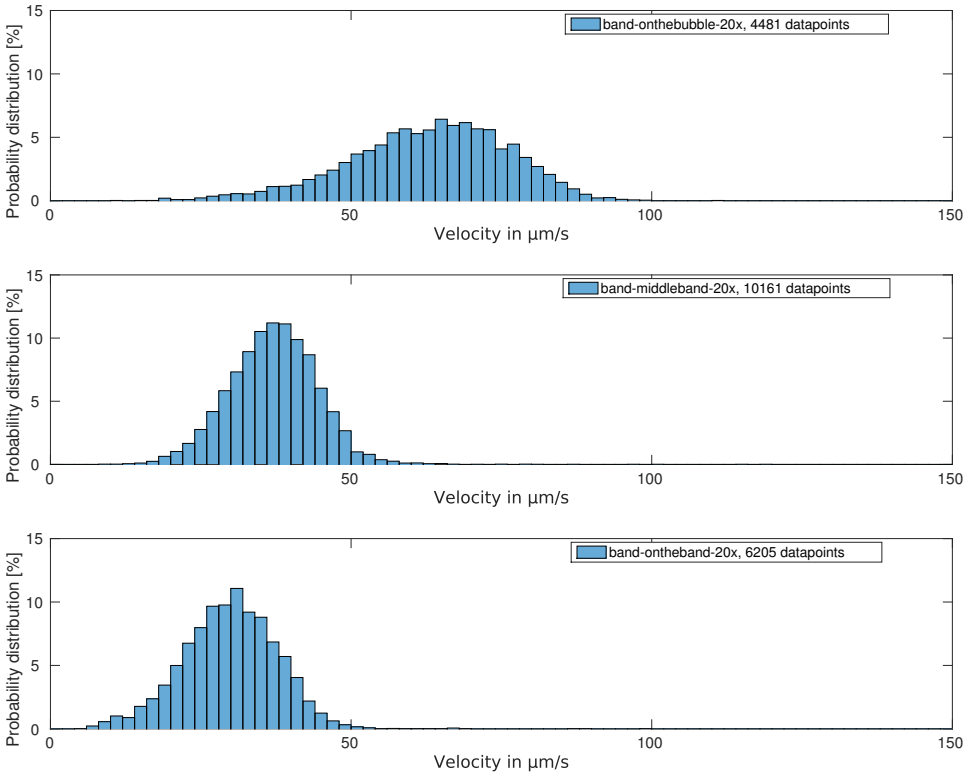


Figure A.10: Velocity distributions for the average of every trajectory.

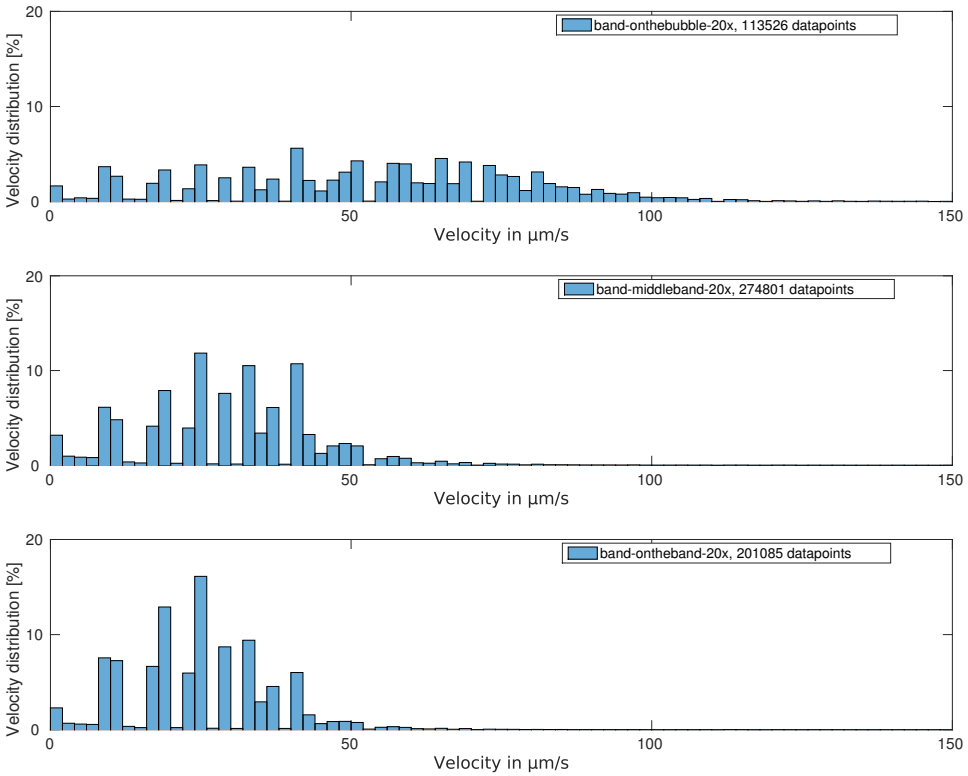


Figure A.11: Velocity distributions for velocities with time steps of 2 frame.

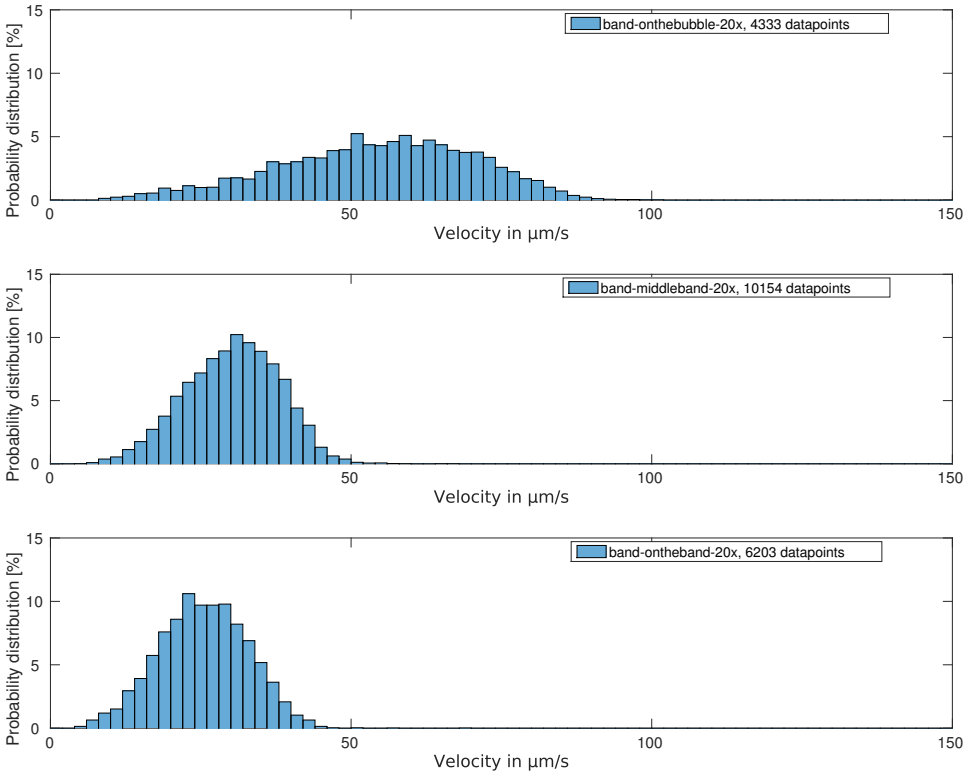


Figure A.12: Average velocity distribution of every trajectory with time steps of 5 frames.

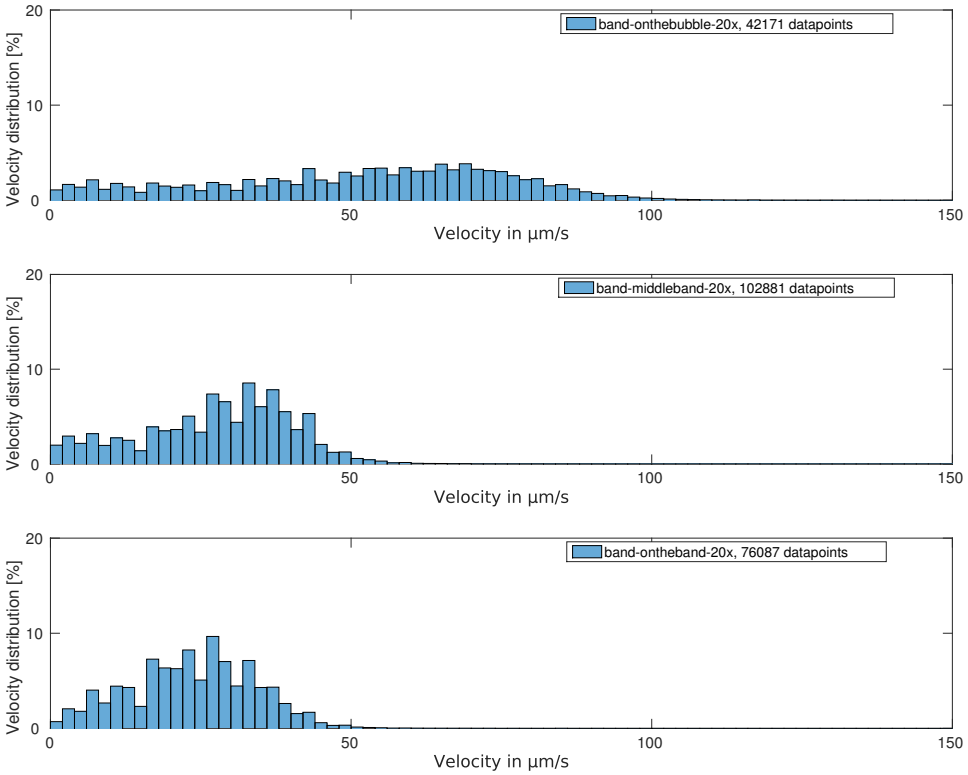


Figure A.13: Velocity distributions for velocities with time steps of 5 frames.

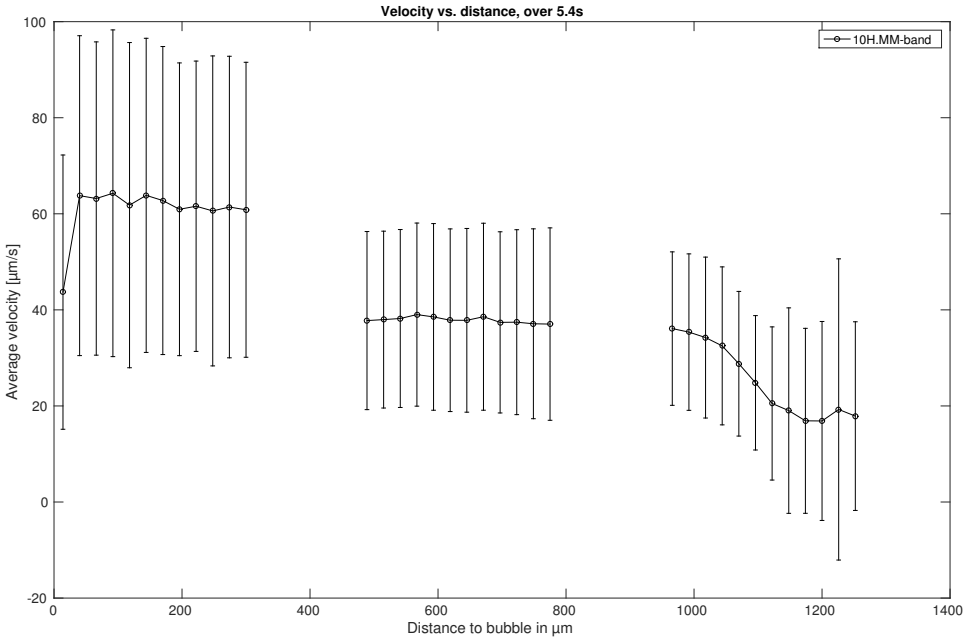


Figure A.14: Velocity profile of the bacterial band. Data are averaged over 5.4s.

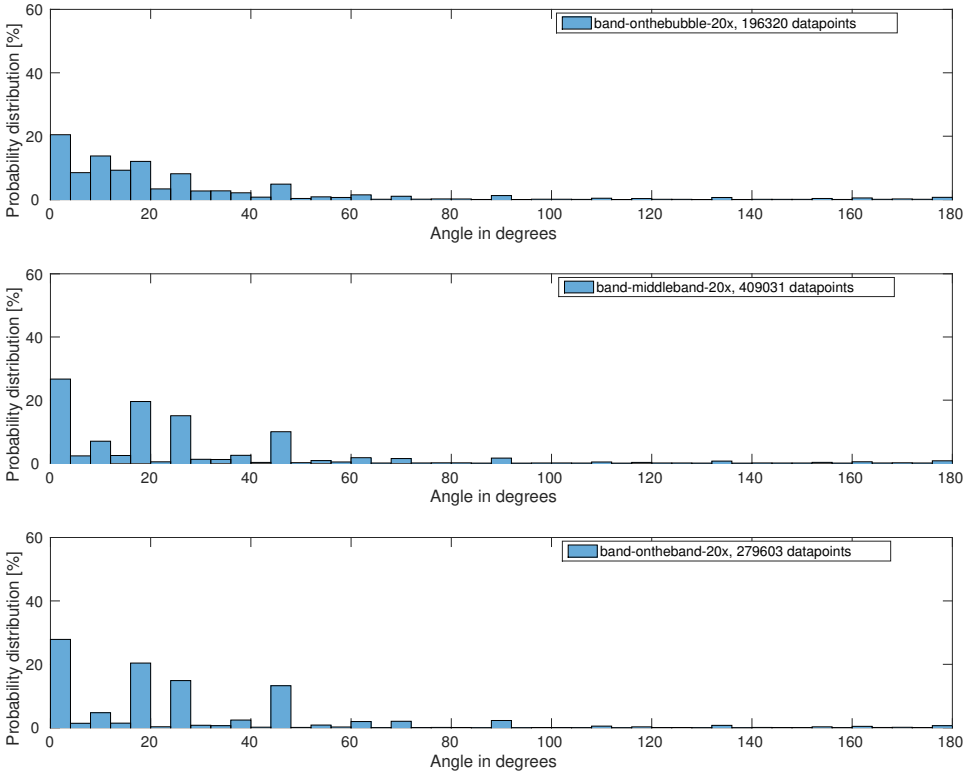


Figure A.15: Probability distribution of the angles between consecutive velocity vectors with minimal distance of $\Delta x = 0.31\,\mu$m.

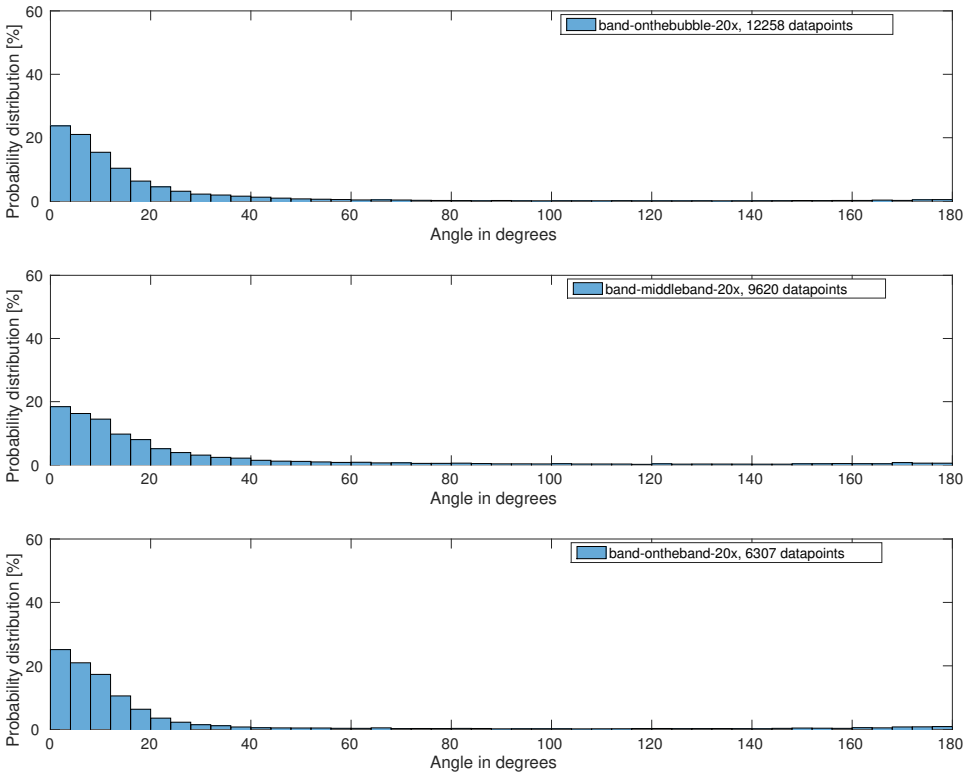


Figure A.16: Probability distribution of the angles between consecutive velocity vectors with minimal distance of $\Delta x = 5.6\,\mu$m.

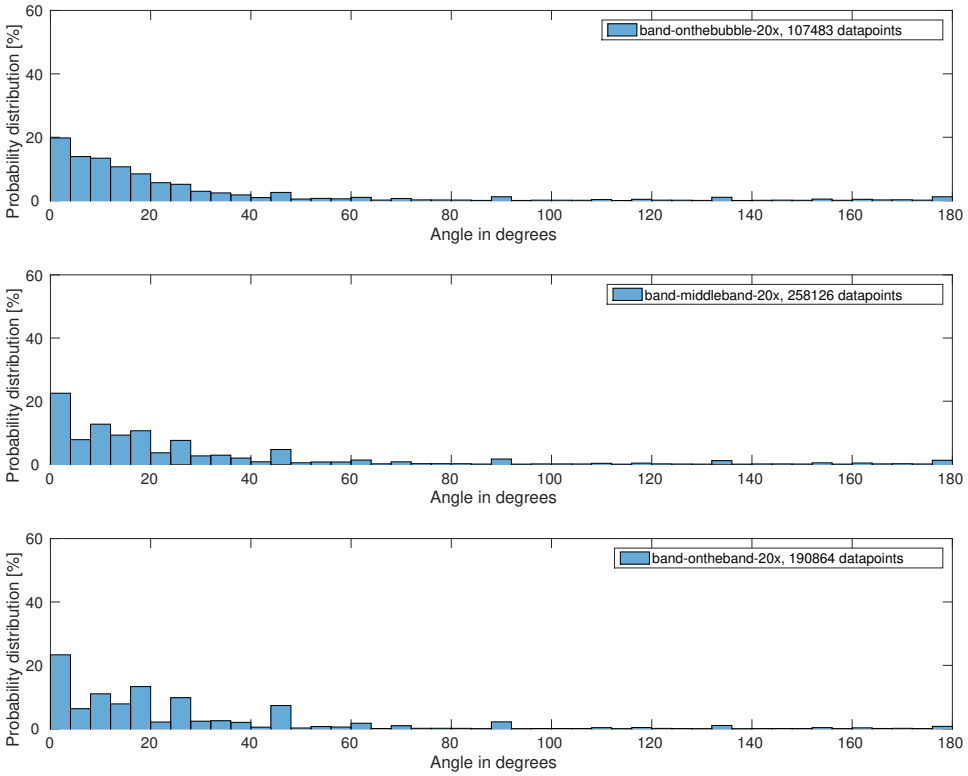


Figure A.17: Probability distribution of the angles between consecutive velocity vectors with a time step of two frame.

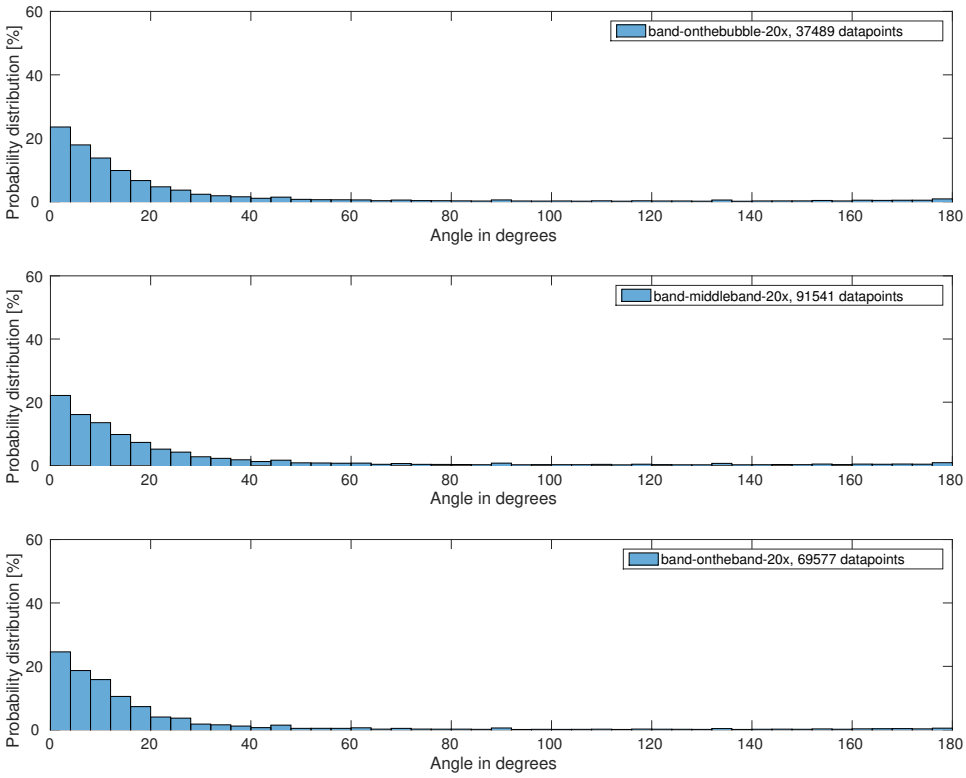


Figure A.18: Probability distribution of the angles between consecutive velocity vectors with a time step of five frames.

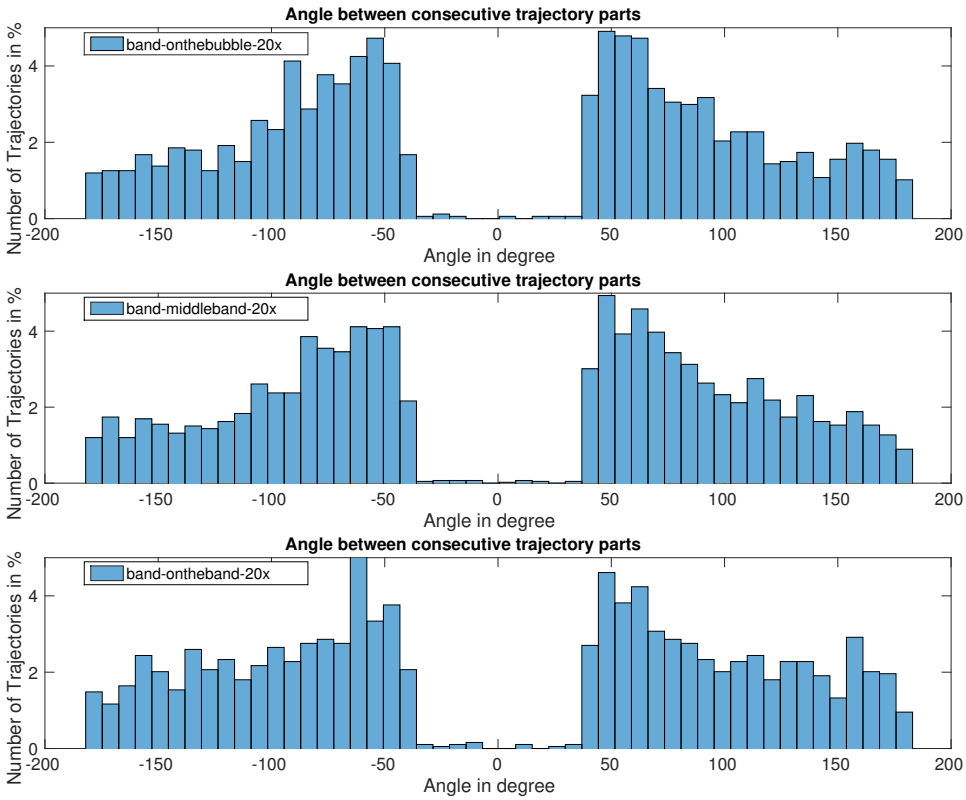


Figure A.19: Probability distribution of the angles between consecutive trajectory segments with $\Delta x = 0.49\,\mu$m. Threshold of 40° for new trajectory segments.

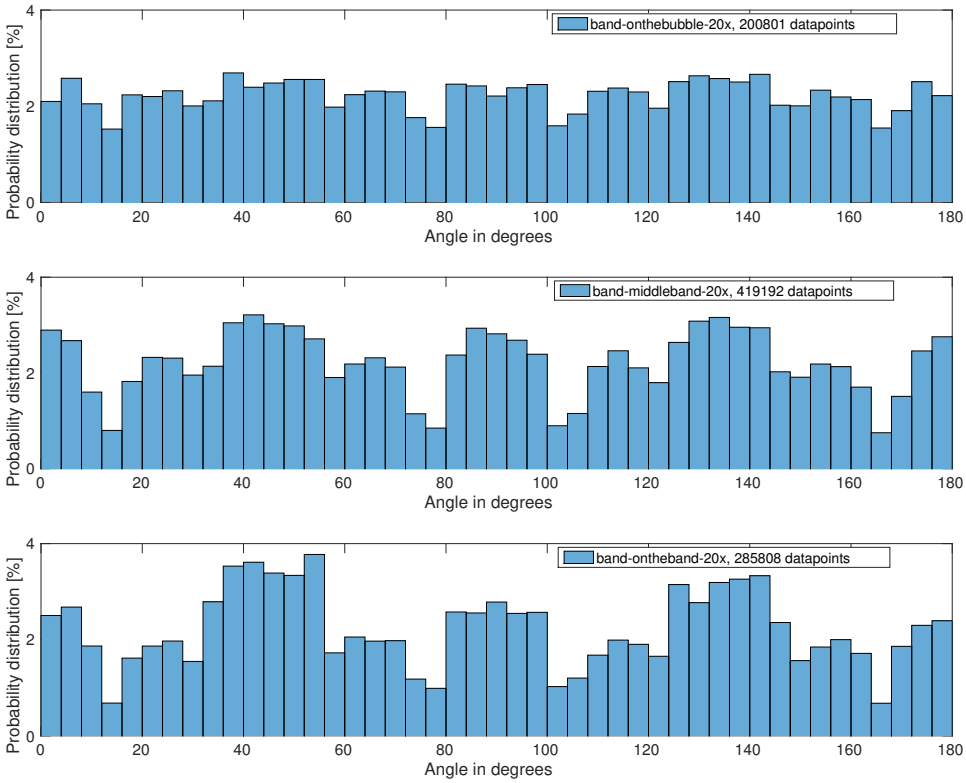


Figure A.20: Probability distribution function of the average relative angle to the bubble inverse normal with minimal velocity of $\Delta x = 0.31\,\mu$m on the position.

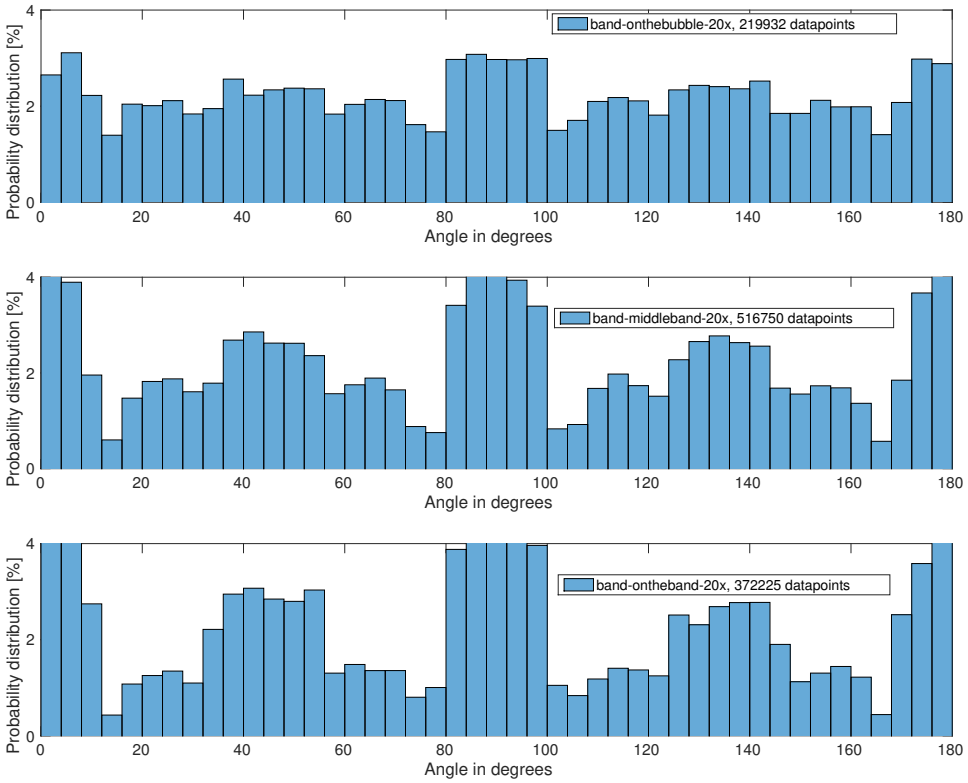


Figure A.21: Probability distribution function of the average relative angle to the bubble inverse normal with time steps of 2 frame.

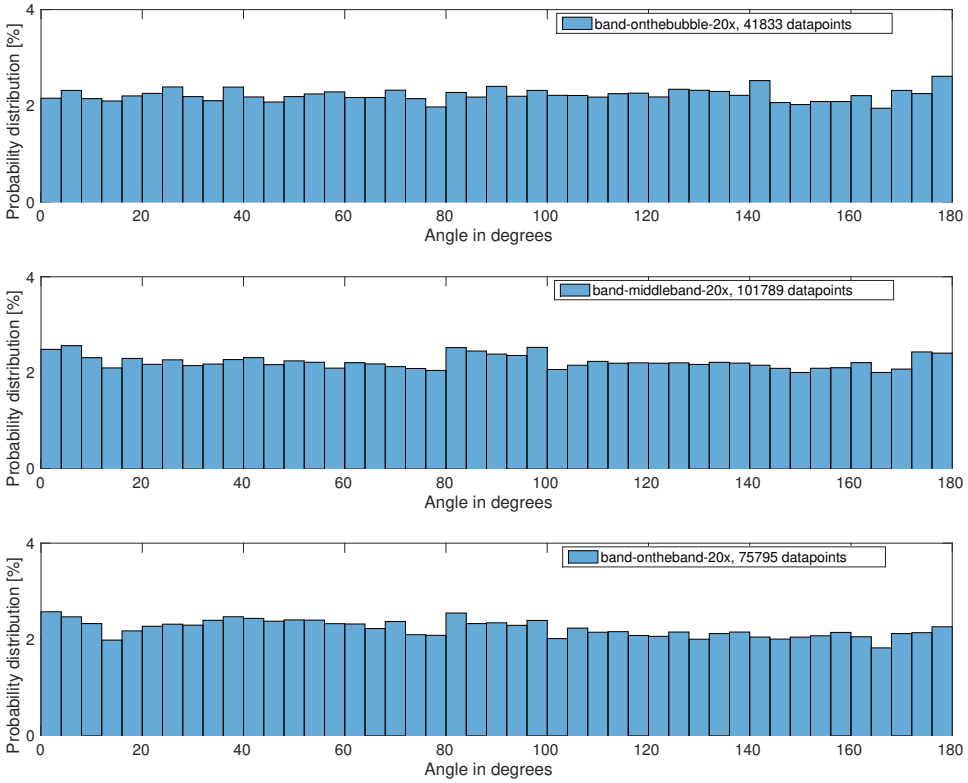


Figure A.22: Probability distribution function of the average relative angle to the bubble inverse normal with with time steps of 5 frames.

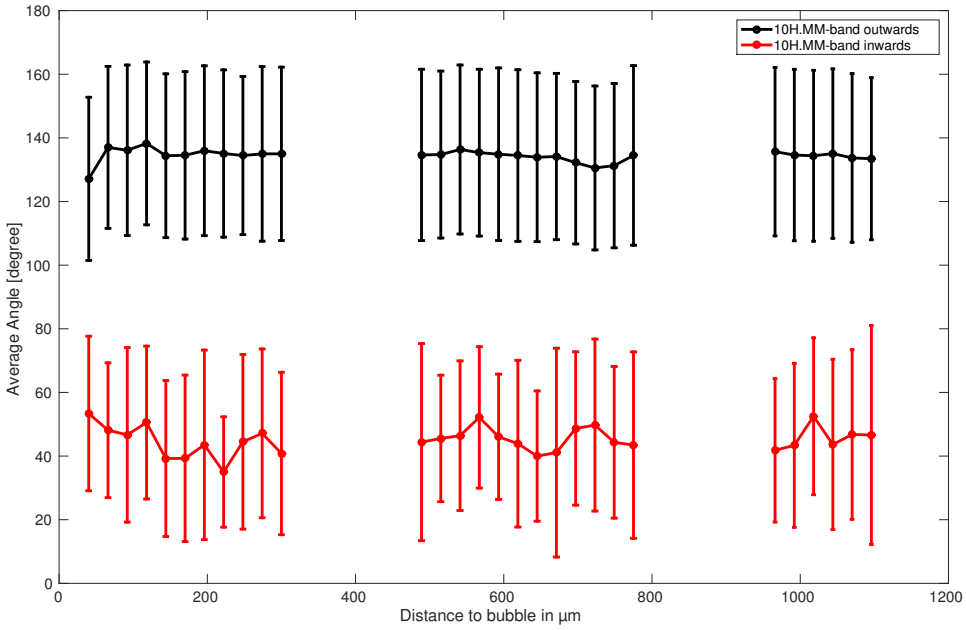


Figure A.23: Dependence of the average relative angle to the bubble inverse normal with minimal velocity of $\Delta x = 5.6\,\mu$m.

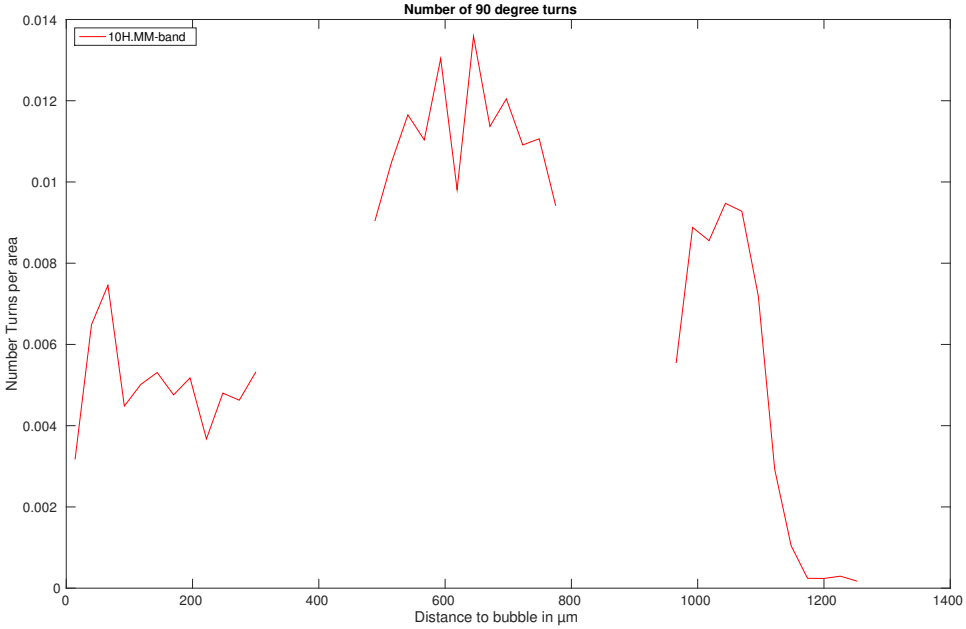


Figure A.24: Number of 90 degree turns between consecutive trajectory segments.

References

[1] H. W. Harris, M. Y. El-Naggar, O. Bretschger, M. J. Ward, M. F. Romine, A. Y. Obraztsova, and K. H. Nealson, *Electrokinesis is a microbial behavior that requires extracellular electron transport*, Proceedings of the National Academy of Sciences USA **107**, 326 (2010).

[2] S. D. Brown, M. R. Thompson, N. C. VerBerkmoes, K. Chourey, M. Shah, J. Zhou, R. L. Hettich, and D. K. Thompson, *Molecular dynamics of theShewanella oneidensisResponse to chromate stress*, Molecular & Cellular Proteomics **5**, 1054 (2006).

[3] M. Vert, Y. Doi, K.-H. Hellwich, M. Hess, P. Hodge, P. Kubisa, M. Rinaudo, and F. Schué, *Terminology for biorelated polymers and applications (IUPAC recommendations 2012)*, Pure and Applied Chemistry **84**, 377 (2012).

[4] L. Hall-Stoodley, J. W. Costerton, and P. Stoodley, *Bacterial biofilms: from the natural environment to infectious diseases*, Nature Reviews Microbiology **2**, 95 (2004).

[5] J. W. Costerton, Z. Lewandowski, D. E. Caldwell, D. R. Korber, and H. M. Lappin-Scott, *Microbial biofilms*, Annual Review of Microbiology **49**, 711 (1995).

[6] K. T. Konstantinidis, M. H. Serres, M. F. Romine, J. L. M. Rodrigues, J. Auchtung, L.-A. McCue, M. S. Lipton, A. Obraztsova, C. S. Giometti, K. H. Nealson, J. K. Fredrickson, and J. M. Tiedje, *Comparative systems biology across an evolutionary gradient within the Shewanella genus*, Proceedings of the National Academy of Sciences **106**, 15909 (2009).

[7] K. Venkateswaran, D. P. Moser, M. E. Dollhopf, D. P. Lies, D. A. Saffarini, B. J. MacGregor, D. B. Ringelberg, D. C. White, M. Nishijima, H. Sano, J. Burghardt, E. Stackebrandt, and K. H. Nealson, *Polyphasic taxonomy of the genus Shewanella and description of Shewanella oneidensis*, International Journal of Systematic Bacteriology **49**, 705 (1999).

[8] C. R. Myers and K. H. Nealson, *Bacterial manganese reduction and growth with manganese oxide as the sole electron acceptor*, Science **240**, 1319 (1988).

[9] G. E. Pinchuk, O. V. Geydebrekht, E. A. Hill, J. L. Reed, A. E. Konopka, A. S. Beliaev, and J. K. Fredrickson, *Pyruvate and lactate metabolism by Shewanella oneidensis MR-1 under fermentation, oxygen limitation, and fumarate respiration conditions*, Applied and Environmental Microbiology **77**, 8234 (2011).

[10] Y. A. Gorby, S. Yanina, J. S. Mclean, K. M. Rosso, D. Moyles, A. Dohnalkova, T. J. Beveridge, I. S. Chang, B. H. Kim, K. S. Kim, D. E. Culley, S. B. Reed, M. F. Romine, D. A. Saffarini, E. A. Hill, L. Shi, D. A. Elias, D. W. Kennedy, G. Pinchuk, K. Watanabe, S. Ishii, B. Logan, K. H. Nealson, and J. K. Fredrickson, *Electrically conductive bacterial nanowires produced by Shewanella oneidensis strain MR-1 and other microorganisms*, Proceedings of the National Academy of Sciences USA **103**, 11358 (2006).

[11] G. Reguera, K. D. McCarthy, T. Mehta, J. S. Nicoll, M. T. Tuominen, and D. R. Lovley, *Extracellular electron transfer via microbial nanowires*, Nature **435**, 1098 (2005).

[12] O. Bretschger, A. Obraztsova, C. A. Sturm, I. S. Chang, Y. A. Gorby, S. B. Reed, D. E. Culley, C. L. Reardon, S. Barua, M. F. Romine, J. Zhou, A. S. Beliaev, R. Bouhenni, D. Saffarini, F. Mansfeld, B.-H. Kim, J. K. Fredrickson, and K. H. Nealson, *Current production and metal oxide reduction by Shewanella oneidensis MR-1 wild type and mutants*, Applied and Environmental Microbiology **73**, 7003 (2007).

[13] K. M. Thormann, R. M. Saville, S. Shukla, D. A. Pelletier, and A. M. Spormann, *Initial phases of biofilm formation in Shewanella oneidensis MR-1*, Journal of Bacteriology **186**, 8096 (2004).

[14] S. T. Read, P. Dutta, P. L. Bond, J. Keller, and K. Rabaey, *Initial development and structure of biofilms on microbial fuel cell anodes*, BMC Microbiology BMC Microbiol **10**, 98 (2010).

[15] H. W. Harris, M. Y. El-Naggar, and K. H. Nealson, *Shewanella oneidensis MR1 chemotaxis proteins and electron-transport chain components essential for congregation near insoluble electron acceptors*, Biochm. Soc. Trans. Biochemical Society Transactions **40**, 1167 (2012).

[16] J. K. Fredrickson, M. F. Romine, A. S. Beliaev, J. M. Auchtung, M. E. Driscoll, T. S. Gardner, K. H. Nealson, A. L. Osterman, G. Pinchuk, J. L. Reed, D. A. Rodionov, J. L. M. Rodrigues, D. A. Saffarini, M. H. Serres, A. M. Spormann, I. B. Zhulin, and J. M. Tiedje, *Towards environmental systems biology of Shewanella*, Nature Reviews Microbiology **6**, 592 (2008).

[17] C. Bücking, F. Popp, S. Kerzenmacher, and J. Gescher, *Involvement and specificity of Shewanella oneidensis outer membrane cytochromes in the reduction of soluble and solid-phase terminal electron acceptors*, FEMS Microbiology Letters **306**, 144 (2010).

[18] E. Marsili, D. B. Baron, I. D. Shikhare, D. Coursolle, J. A. Gralnick, and D. R. Bond, *Shewanella secretes flavins that mediate extracellular electron transfer*, Proceedings of the National Academy of Sciences USA **105**, 3968 (2008).

[19] S. Pirbadian, S. E. Barchinger, K. M. Leung, H. S. Byun, Y. Jangir, R. A. Bouhenni, S. B. Reed, M. F. Romine, D. A. Saffarini, L. Shi, Y. A. Gorby, J. H. Golbeck, and M. Y. El-Naggar, *Shewanella oneidensis MR-1 nanowires are outer membrane and periplasmic extensions of the extracellular electron transport components*, Proceedings of the National Academy of Sciences USA **111**, 12883 (2014).

[20] M. Y. El-Naggar, G. Wanger, K. M. Leung, T. D. Yuzvinsky, G. Southam, J. Yang, W. M. Lau, K. H. Nealson, and Y. A. Gorby, *Electrical transport along bacterial nanowires from Shewanella oneidensis MR-1*, Proceedings of the National Academy of Sciences USA **107**, 18127 (2010).

[21] A. Korenevsky and T. J. Beveridge, *The surface physicochemistry and adhesiveness of Shewanella are affected by their surface polysaccharides*, Microbiology **153**, 1872 (2007).

[22] M. J. Marshall, A. S. Beliaev, A. C. Dohnalkova, D. W. Kennedy, L. Shi, Z. Wang, M. I. Boyanov, B. Lai, K. M. Kemner, J. S. Mclean, S. B. Reed, D. E. Culley, V. L. Bailey, C. J. Simonson, D. A. Saffarini, M. F. Romine, J. M. Zachara, and J. K. Fredrickson, *C-type cytochrome-dependent formation of U(IV) nanoparticles by Shewanella oneidensis*, PLoS Biol **4**, 268 (2006).

[23] L. Xie, T. Altindal, S. Chattopadhyay, and X.-L. Wu, *Bacterial flagellum as a propeller and as a rudder for efficient chemotaxis*, Proceedings of the National Academy of Sciences USA **108**, 2246 (2011).

[24] A. Paulick, A. Koerdt, J. Lassak, S. Huntley, I. Wilms, F. Narberhaus, and K. M. Thormann, *Two different stator systems drive a single polar flagellum in Shewanella oneidensis MR-1*, Molecular Microbiology **71**, 836 (2009).

[25] K. M. Thormann and A. Paulick, *Tuning the flagellar motor*, Microbiology **156**, 1275 (2010).

[26] J. Schindelin, I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, S. Preibisch, C. Rueden, S. Saalfeld, B. Schmid, J.-Y. Tinevez, D. J. White, V. Hartenstein, K. Eliceiri, P. Tomancak, and A. Cardona, *Fiji: an open-source platform for biological-image analysis*, Nature Methods **9**, 676 EP , perspective (2012).

[27] C. T. Rueden, J. Schindelin, M. C. Hiner, B. E. DeZonia, A. E. Walter, E. T. Arena, and K. W. Eliceiri, *Imagej2: Imagej for the next generation of scientific image data*, BMC Bioinformatics **18**, 529 (2017).

[28] P. J. Burt, *Fast filter transform for image processing*, Computer Graphics and Image Processing **16**, 20 (1981).

[29] D. Reid, *An algorithm for tracking multiple targets*, IEEE Transactions on Automatic Control **24**, 843–854 (1979).

[30] K. Jaqaman, D. Loerke, M. Mettlen, H. Kuwata, S. Grinstein, S. L. Schmid, and G. Danuser, *Robust single-particle tracking in live-cell time-lapse sequences*, Nature Methods **5**, 695 (2008).

[31] D. Chetverikov and J. Verestói, *Feature point tracking for incomplete trajectories*, Computing **62**, 321 (1999).

[32] R. Jonker and A. Volgenant, *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, Computing **38**, 325 (1987).

[33] R. E. Kalman, *A new approach to linear filtering and prediction problems*, Transactions of the ASME–Journal of Basic Engineering **82**, 35 (1960).

Erklärung

Nach §18(8) der Prüfungsordnung für den Bachelor-Studiengang Physik an der Universität Göttingen:
Ich versichere hiermit, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die von mir angegebenen Quellen und Hilfsmittel verwendet habe. Wörtlich oder sinngemäß aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht. Die Richtlinien zur Sicherung der guten wissenschaftlichen Praxis an der Universität Göttingen wurden von mir beachtet. Eine gegebenenfalls eingereichte digitale Version stimmt mit der schriftlichen Fassung überein. Mir ist bewusst, dass bei Verstoß gegen diese Grundsätze die Prüfung mit nicht bestanden bewertet wird.

Göttingen, July 16, 2018

Thomas Breithaupt