Improving bright-field images for tracking particles with Defocustracker

Bachelor Project in Biomedical Engineering
Department of Biomedical engineering
Lund University Faculty of Engineering

Student: Carl Lundin

Supervisor: Thierry Baasch

Examiner: Thomas Laurell

Abstract

Acoustophoresis is a cell separation technology where particles or cells are suspended in a fluid container and subject to acoustic standing waves. The standing waves exerts an acoustic radiation force on the particles, pushing them towards the nodal plane. This movement is called migration, and the migration velocity depends on the acoustic mobility of the particle, which depend on the particle size, density and compressibility relative to the suspension fluid. Acoustophoresis is used in biomedical and clinical applications, such as diagnostics. The velocities of the particles migrating in the container are determined by individually tracking the particles in a software called Defocustracker. A series of images is taken of the migration and are then analyzed in Defocustracker. In order to perform the particle tracking, the images must be dark-field and the particles fluorescent - in bright-field mode Defocustracker cannot properly differentiate between particles and noise in the images, as bright-field introduces a lot of background noise. There is a concern that dying the particles may affect their properties, which is why it would be preferable to conduct the acoustophoresis in bright-field. In some cases, the cells are not even possible to dye, which is another motivation to track cells in bright-field. The aim of this candidate thesis is to find a pre-processing algorithm that is able to improve the images, so that the particle tracking can be done in bright-field.

Acknowledgements
Thanks to Thierry Baasch, you were a great supervisor
and I learnt a lot.
Thanks also to Thomas Laurell.

Contents

1 Background

In biomedicine, cell and particle separation has led to many innovations such as efficiently isolating penicillin from cell culture compounds[1]. Over the years, many different techniques and methods have been developed. Acoustophoresis is such cell separation technique, and is an important and versatile method in cell and particle separation[2]. In diagnostics, for example, cell separation can be used to determine the concentration of cellular bio-markers in a blood sample[3]. In purification processes, it can be used to filter out particles from a complex compound, also known as 'washing'[4] or increase the concentration of cells in a fluid, known as 'enrichment'[3]. In micro-fluidics, the flow movement of the liquid is called laminar flow. At this scale, the fluids move smoothly in layers, and are not mixing as in a turbulent flow. It is possible to predict if a flow is turbulent or laminar using Reynolds number, which is determined by the viscosity, flow velocity and size of the fluid container[5]. At the micro-scale, the small container size renders a small Reynolds number, indicating a laminar flow. The Reynolds number is defined by:

$$Re = \frac{\rho u l}{\mu} \tag{1}$$

where ρ is the density of the fluid, u is the flow speed, l is the hydraulic diameter of the body of the fluid and μ is the kinematic viscosity of the fluid. The hydraulic diameter is given by $d_h = \frac{4A}{p}$, where A is the cross-section area of the fluid container, and p is the wetted perimeter of the container.

In acoustophoresis, cells are suspended in a micro-fluid container which is subject to acoustic sinusoid waves. The acoustic waves create standing waves in the micro-fluid container, with a pressure node in the center area. These standing waves exerts a radiation pressure on the cells and pushes them towards the pressure node, if they have a positive contrast factor. Particles with a negative contrast factor, such as lipid particles, are pushed to the anti-node at the channel side walls[6]. The acoustic radiation force acting upon a particle is given by the equation:

$$F_{rad} = 4\pi\Phi\, a^3 k E_{ac} sin(2ky) \tag{2}$$

E_{ac} is the acoustic energy density, and is a measure of the acoustic intensity per unit volume. Different fluid compounds in different micro-fluid channels renders different acoustic energy densities[7]. Φ is the contrast factor, which is a measure of the particle density and mobility relative to the surrounding medium. It is calculated using reference particle values. a is the radius of the particle, k is the wave number and y is the position of the particle relative to the channel center.

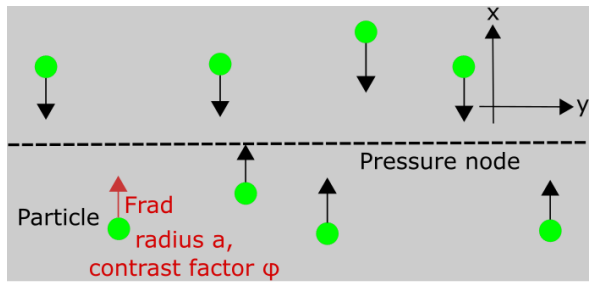


Figure 1: The acoustic radiation force F_{rad}, the contrast factor Φ and the particle radius a.

As seen in figure 2, the force exerted by a horizontal standing wave pushes the particles to the same z-plane, levitation, and the force exerted by a transversal standing wave pushes the particle to migrate towards the pressure node in the center area of the channel, migration. The migration movement is what is being recorded by the camera in this thesis project.
The acoustic mobility of a particle is proportional to the migrational speed in an acoustic manipulation channel, and the acoustic migration behaviour of particles or cells reveals much about their acousto-mechanical properties.

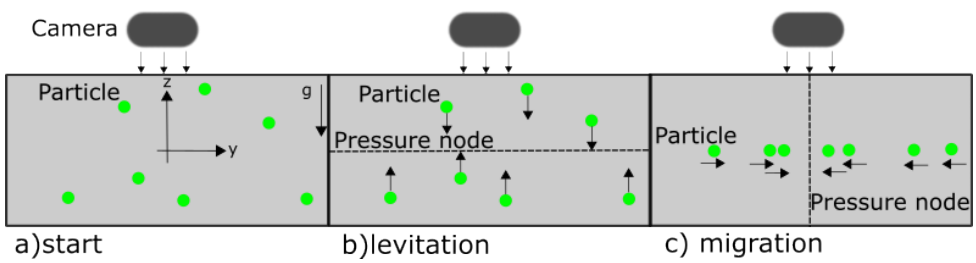


Figure 2: The horizontal and transversal standing waves in the micro-channel exerts an acoustic radiation force on the particles, causing them to levitate in the same z-plane, and then migrate towards the center area of the channel.

2 Motivation

2.1 The goal of the project

When performing acoustophoresis described in the previous section, photographs are taken of the particles as they migrate towards the pressure node. The image series resulting from the photographs are processed with a software called Defocustracker[9]. Experiments are currently being done using fluorescent particles in dark-field microscopy. However, when handling cells or particles, it is preferable to handle them as little as possible. The process of dying cells may have an impact on the cell sample, and by performing the tracking in bright-field, we can thus eliminate a step when handling the particles. The problem is that bright-field microscopy introduces noise in the images, and the particles can not be tracked in Defocustracker software in a straight forward way. The overall goal of this project is to

find ways to pre-process the image sets to reduce the noise, and locate the particle signals in the images, so that the particles can be tracked properly with Defocustracker.

2.2 The experiment

In this thesis project at BMC, Lund, under the direction of professor Thomas Laurell, particles or cells are suspended in a liquid inside a micro-channel fabricated in silicon and covered by a glass lid. Two piezoelectric transducers are connected to the micro-channel, and the oscillating voltage signals are transformed into a mechanical vibration, creating the standing waves, known as 'bulk standing waves' (BAW)[1]. The standing waves are both horizontal and transversal, as shown in figure 2. The smaller transducer with the shorter wavelength, 5 MHz, creates the horizontal standing wave, and the larger transducer with 2 MHz wavelength creates the transversal standing wave. The resonance frequency of a transducer depends on its thickness[7].

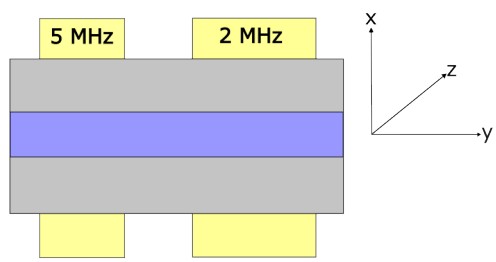


Figure 3: The acoustophoresis chip with the microfluid container and the transducers, which are actuated by 5 and 2 MHz, respectively. The transducers create bulk standing waves within the container. The particles will be located at the same depth (z-plane) in the channel.

The acoustic radiation force causes the particles to migrate towards the pressure node in the center of the micro fluid container, and their velocity profiles are sinusoidal. Above the container, a microscope camera is located, taking multiple images during the cell migration, illustrated in figure 4. With the image series the particles can be tracked, and their velocity determined.

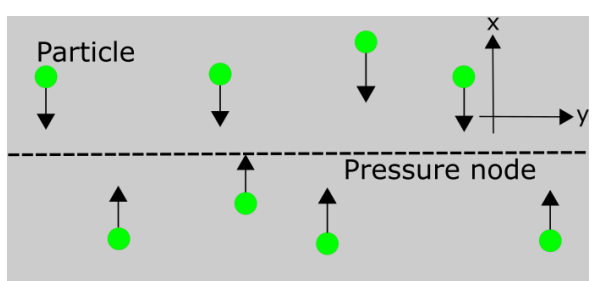


Figure 4: Top-view of the particle migration towards the pressure node (dotted line) in the center area of the micro-channel.

The tracking is done with the Defocustracker software.[8] Defocustracker utilizes a principle in particle tracking, called 'General Defocusing Particle Tracking' (GDPT), and can track particles in 3 dimensions, by determining the amount of defocusing in the image, since this reveals how close a particle is to the focal plane. In this particular experiment measurements in z-direction is not of interest, since the particles are focused into a single plane - levitation - before the migration. Thus, we use Defocustracker in 2D mode. In this mode, one image from the image set is chosen as a calibration image. In this frame, a single particle is chosen to serve as calibration reference when tracking other particles in the image set. With this reference particle, a normalized cross-correlation is performed on each frame in the set and a correlation map with the particle center positions is created. The correlation map contains values between 0 and 1 with their peak values in the particle center, where shapes similar to the calibration particle are found[9]. In short, similarities between regions in the each of the frames and the calibration particle are evaluated using cross-correlation.

There are different tracking algorithms available in Defocustracker. We are using the 'nearest-neighbor'-algorithm in which the program compares the coordinates of the found particles in one image with the particle coordinates from the subsequent image, pairing the nearest particle coordinates. Other tracking algorithms available in Defocustracker pairs the particles based on ratio of similarity, or uses a non-linear scale when pairing the particles. The 'nearest-neighbor'-algorithm is the default setting in Defocustracker, and fits the purposes of this experiment, where we do want to pair particles between frames based on their proximity to one another.

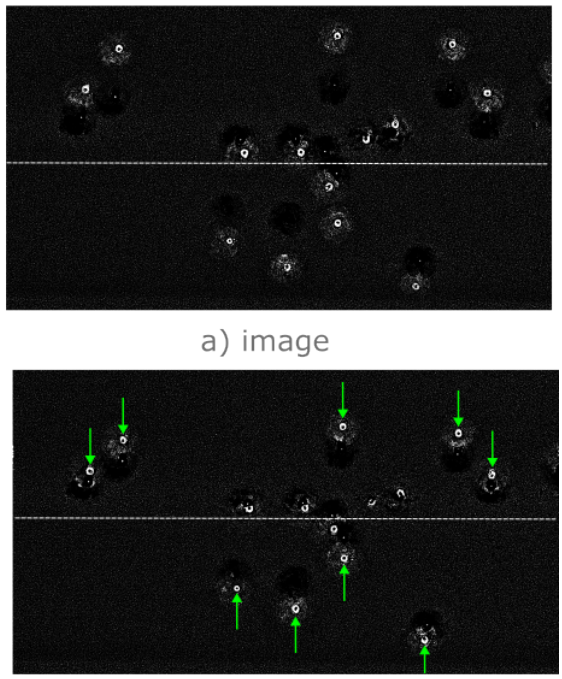


Figure 5: How the 'nearest-neighbor'-algorithm works. Defocustracker identifies particles closest to the positions in the subsequent frame. The nodal plane is represented by the the dashed line.

In this way, all the position and velocity data is collected which are used to compute trajectories of the individual particles from an image set.

2.3 The bright-field issue

The difference between dark-field and bright-field images can be seen in figure 6 and 7.

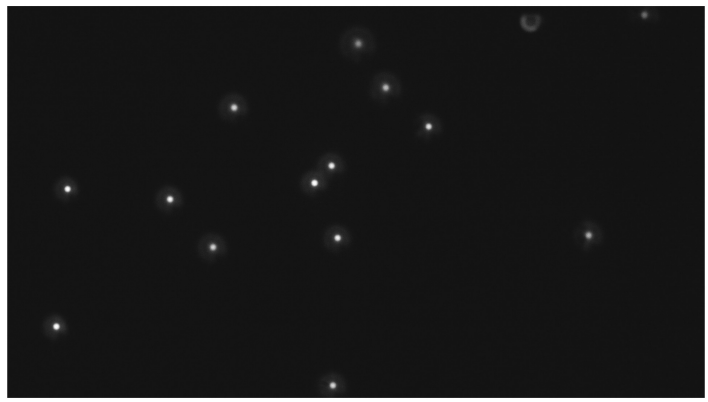


Figure 6: Fluorescent particles in darkfield.

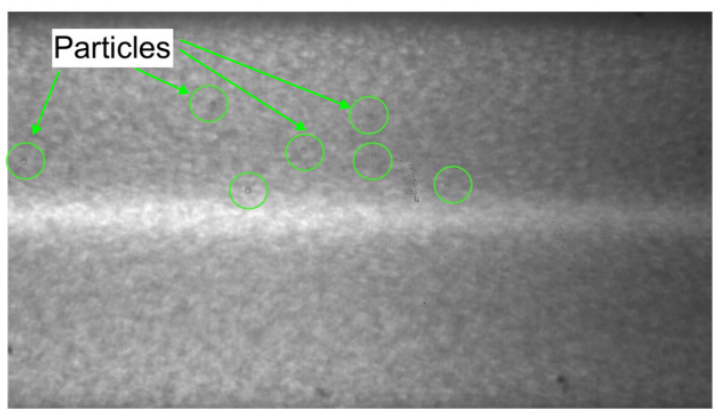


Figure 7: Particles in brightfield. Some particles are circled in green.

As can be seen in figure 7, bright-field mode introduces a lot of background noise. The particles are not easy to locate, even by eye. The Defocustracker software is not able to properly differentiate between the particle signals and the noise in the images - the noise is too similar to the particles.

As stated in the beginning of this section, the goal of the project is to find a pre-processing algorithm that improves the images so that the particles can be tracked in bright-field with Defocustracker software.

3 Methods

3.1 Initial ideas and approach

In order to maintain a consistent work-flow, it was preferable to process the images using Matlab code, since most other code related to the particle tracking in the lab environment is written in Matlab. Rather than adding to or modifying the Defocustracker script, we wanted to find a pre-processing step in which the images series can be processed with the Defocustracker software.

The first idea was to use background subtraction to get rid of background noise. The last image of a particular set was subtracted from the other images in the same set, because in the images used in this project, the particles in the last image of the sets were all located at the nodal center and had zero velocity. This way, no particles of interest would be cancelled out in the subtraction, and the background of the last image would be similar to the background in the rest of the images. Since the noise from the bright-field background has random elements - noise from electrical circuits, variations in background light - it is impossible to extract background noise completely using this technique. The result of background subtraction can be seen in figure 8.

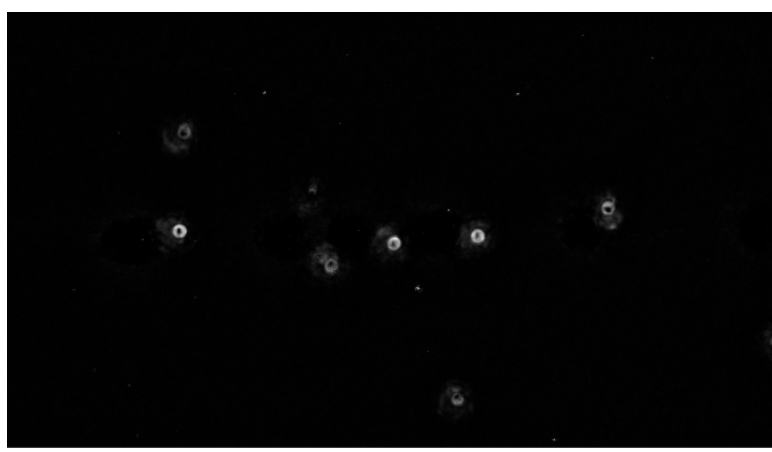


Figure 8: Particles in brightfield. Resulting image after background subtraction, using an image from the same image set for background subtraction.

In the original image, figure 9, there is a bright dot in the center of the particles. This is the reflection from the top - we are looking at a spherical object from above.

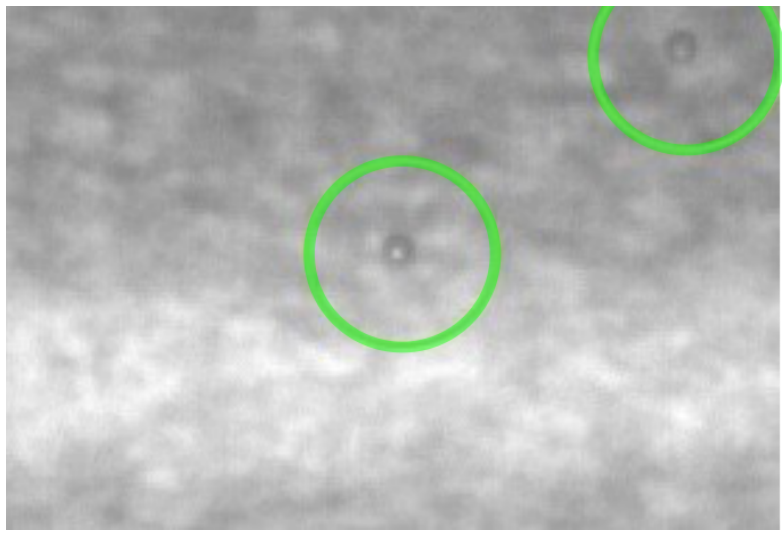


Figure 9: Original bright field image. The bright dot in the middle is light reflected off the top of the particle - seen from above it is a dome-shaped object.

After subtraction this dot is inverted and instead there is a dark spot in the middle of the particles, and bright noise around it. The bright noise around the particles is the inverted shadow from the particles. The camera is located above the micro fluid channel, and the light shone from above cast shadows in the channel, figure 11.
These shadows creates some challenges when tracking and detecting the particles - the shadows were registered as particles. There were also background noise not originating from the particle shadows, but from background light and electrical circuits, as mentioned above.

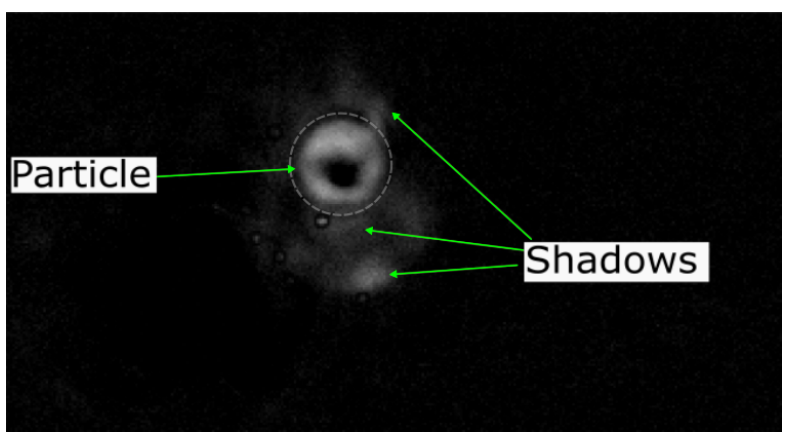


Figure 10: Particle and shadows after background subraction.

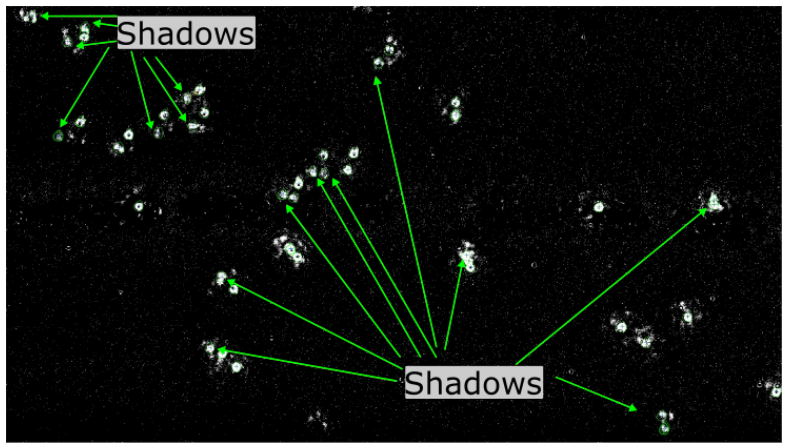


Figure 11: Particle and shadows, after background subtraction. Shadows were sometimes identified as particles.

The next idea was to amplify the image signal, and then add a threshold and cut out all the signals under the threshold value. That way the stronger signals would be preserved, and the noise filtered out. The problem with this approach was that shadows were amplified as well, and it was a difficult task to set the amount of amplification and threshold value at the proper levels for each of the image series. If the threshold was set too high, both the particle signals and the shadow signals were cancelled out. The amplification and threshold thus had to be fine-tuned for each set. An example of shadows being amplified along with the particle signals can be seen in figure 12. The images were now close to binary, with pixels being at either maximum intensity or zero intensity - the detail and contrast in the images was lost.

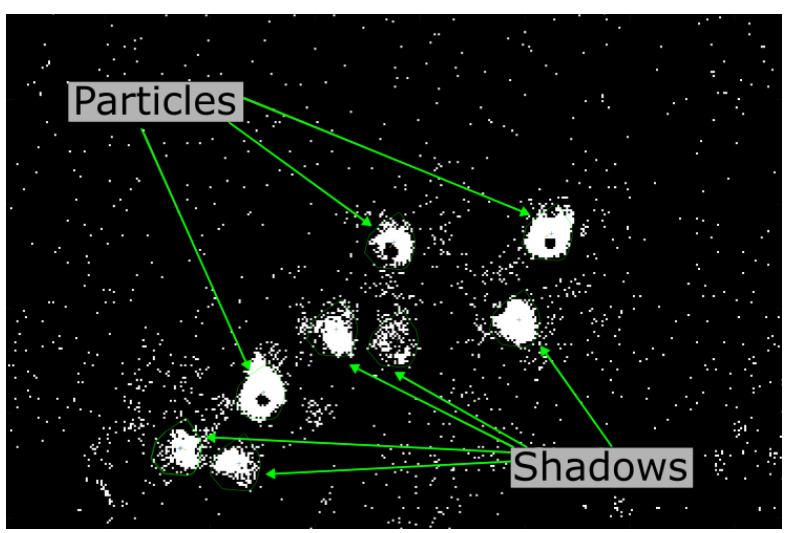
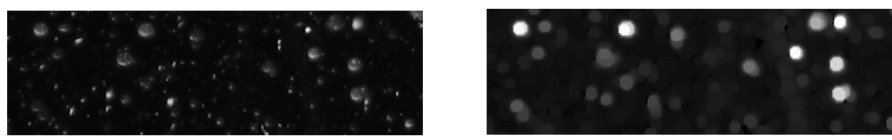


Figure 12: Particle and shadows, after background subtraction and thresholding.

3.2 Matlab image filter to add contrast

Since the Defocustracker used cross-correlation, giving a value between 0 and 1 when comparing images to the calibration particle, the next approach was to improve the contrast and detail in the images, to somehow amplify the edges in the images, facilitating the cross-correlation algorithm in Defocustracker, which identifies similarity in image regions, described earlier. There was a Matlab function called imopen, which morphologically opens an image[10]. It is called in the following manner: openImage = imopen(image, strel) where the structuring element, strel, is defined as:
strel = strel('disk', r) The function takes an image, and in this case removes 'disk'-shaped objects in the image having a radius of r pixels or less.



(a) (b)

Figure 13: Example images from Matlab[10]. To the right: before proccesing with imopen(I, strel). To the left: after processing with imopen(I, strel)

The function 'imopen' consists of two operations: erosion and dilation, performed in series. Erosion takes two inputs: the image being operated on and a structuring element, also known as kernel. This element is in our case a predefined 'disk' and it determines the effect of the operation on the input image. The 'disk' kernel is superimposed on the image matrix, and if radius of the 'disk' is 5, then all the bright pixels that are not surrounded by bright pixels in a neighborhood of 5 pixels, will be set to dark (zero). The dilation is the dual of erosion, i.e. when the 'disk' is superimposed on the image, all the dark pixels that are not surrounded by dark pixels, will be set to bright. The basic principle of erosion and dilation is shown in figure 14.

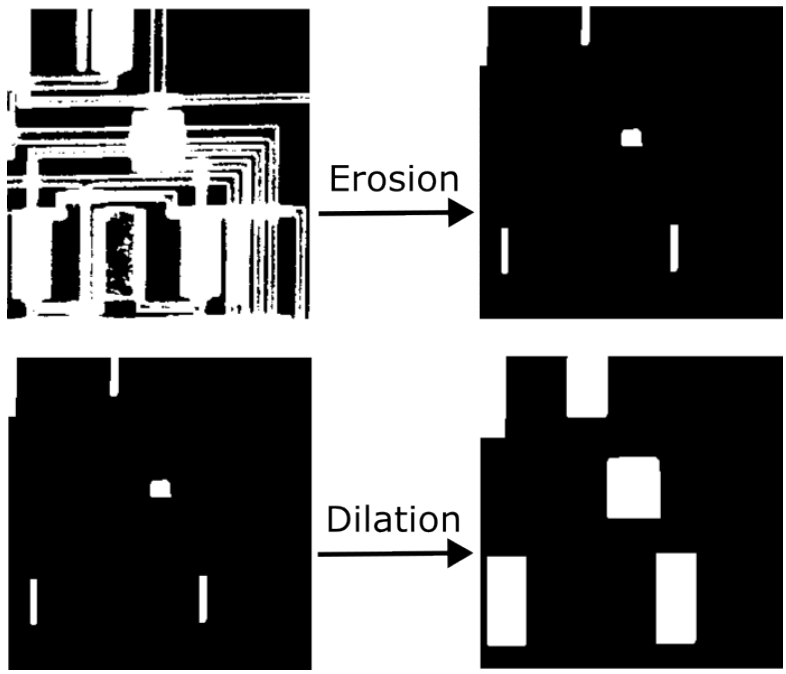


Figure 14: Basic principle of erosion and dilation, here with a rectangular structuring element. The original top-left image is first eroded, and the same eroded image is then dilated.

Applying imopen(I, strel) to the images - both the image being processed and the reference image from the same image series used to subtract the background - and subtracting these two images, creates a filtered background image in which only the larger elements are present.
Subtracting this filtered image from the background subtracted image results in a more detailed image.

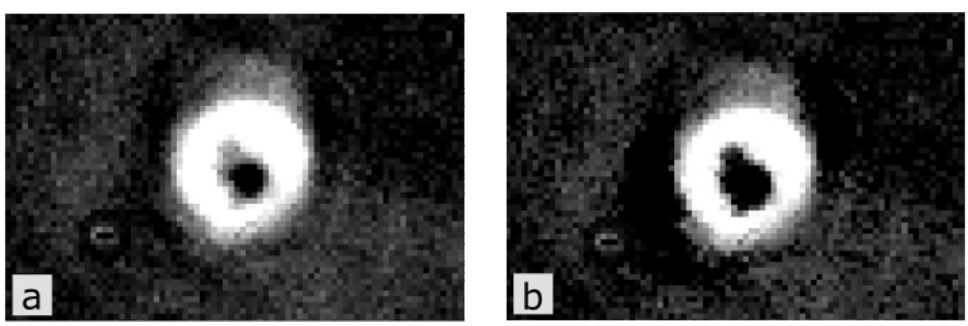


Figure 15: To the left, only background subtraction is used. To the right, the left image has been background subtracted and filtered with imopen(I, strel).

The difference might be subtle, but it is visible that more detail and contrast is added after

this process. However, there were still noise, mainly from the particle shadows, that were registered as particles in Defocustracker (see Results section).

3.3 Laplacian of Gaussian operator

We kept searching for other ways to improve the image processing, mainly in filtering and more subtraction and thresholding techniques. We saw that after running the images through a highpass filter, the desired detail and contrast was there, but it didn't work well with the software.

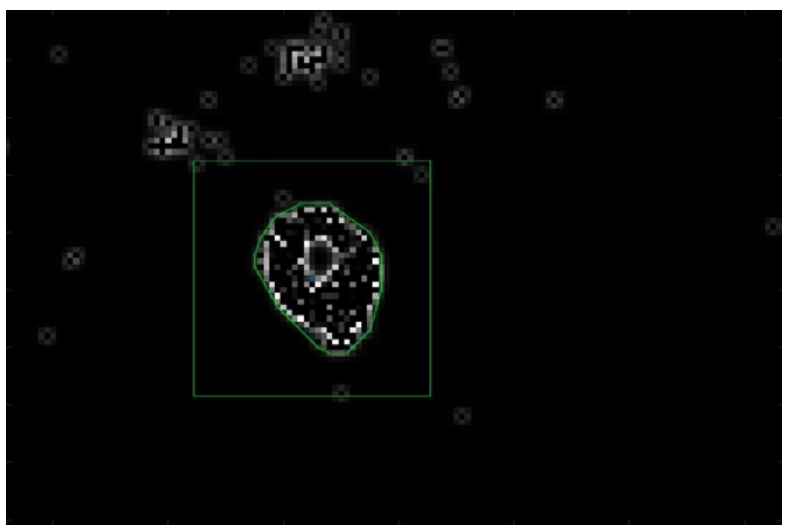


Figure 16: Image of particle after high-pass-filtering.

But the GDTP script was unable to detect the particles.

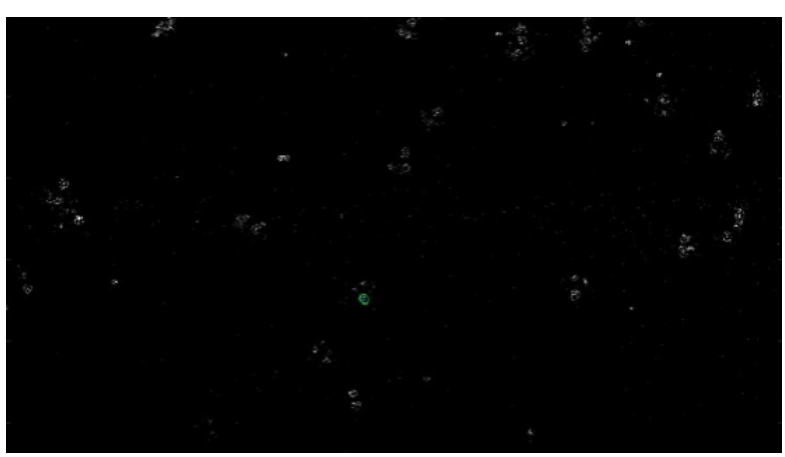


Figure 17: Particle tracking did not work with high-pass-filtered images, only a few of the particles were registered in Defocustracker.

The next idea was to apply a so-called 'mexican hat' filter, a tool often used in edge detection[11].

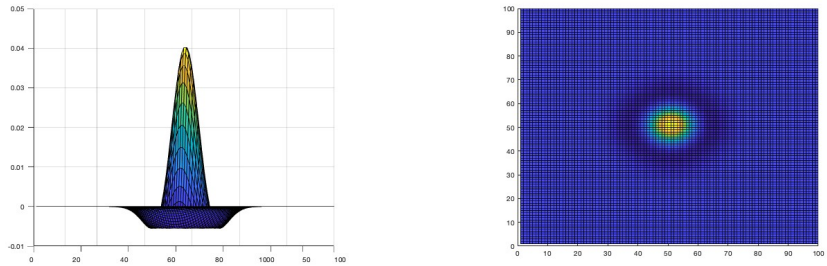


Figure 18: The 'mexican hat' filter characteristics. Left: The principal shape of the 'mexican hat' filter. Right: 2D surface plot of the filter. It is the 'hat' seen from above.

In technical terms, it is called a Laplacian of Gaussian, or LoG filter.

The Gaussian filter is also known as average filter, or 'blur' filter/effect, since Gaussian filter is a low-pass filter where high-frequency regions are averaged, resulting in smoother shapes. [11]

Laplacian of Gaussian, on the other hand, is often used in edge detection, since it accentuates regions of rapid intensity change[11]. Laplacian is the second derivative operation and gives a clear localization of the edges, as the LoG response will be positive on the darker side of an edge, and negative on the brighter side of an edge.

The Gaussian distribution has the mathematical formula:

$$G(x,y) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

Then the Laplacian of Gaussian (2nd derivative of Gaussian) becomes:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}[1 - \frac{x^2+y^2}{2\sigma^2}] \exp^{\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

An illustration of a Laplacian of Gaussian response to a unit step is seen in figure 19.

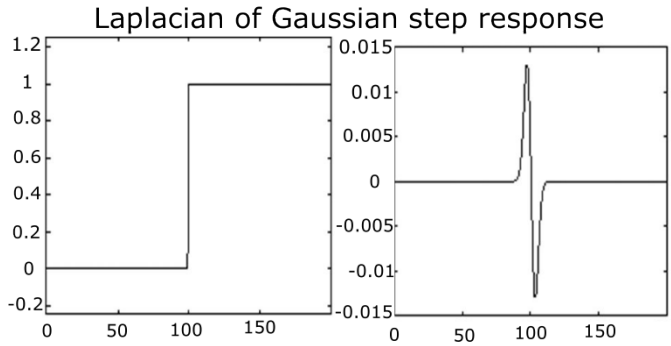


Figure 19: To the left is a step function of a 200 pixel sequence.
To the right is the step response of a Laplacian of Gaussian filter.

This function can also be approximated as a matrix, with σ=1.4, x, y=9 we obtain the matrix in figure 20. When processing images with such a matrix, the matrix is called 'kernel' or 'structuring element' (strel).

0	0	-1	-2	-2	-2	-1	0	0
0	-1	-3	-5	-5	-5	-3	-1	0
-1	-3	-5	-3	0	-3	-5	-3	-1
-2	-5	-3	12	23	12	-3	-5	-2
-2	-5	0	23	40	23	0	-5	-2
-2	-5	-3	12	23	12	-3	-5	-2
-1	-3	-5	-3	0	-3	-5	-3	-1
0	-1	-3	-5	-5	-5	-3	-1	0
0	0	-1	-2	-2	-2	-1	0	0

Figure 20: The filter used in the pre-processing - a discrete approximation of a Laplacian of Gaussian filter.

The kernel is slid across the whole image matrix. In this case a kernel of 9 by 9 pixels is slid across an image of 1084 by 1920 pixels. At each pixel the convolution of the kernel and image is computed. The principle of this operation is illustrated in figure 21, where as an example the convolution is computed at pixel P_{76}.

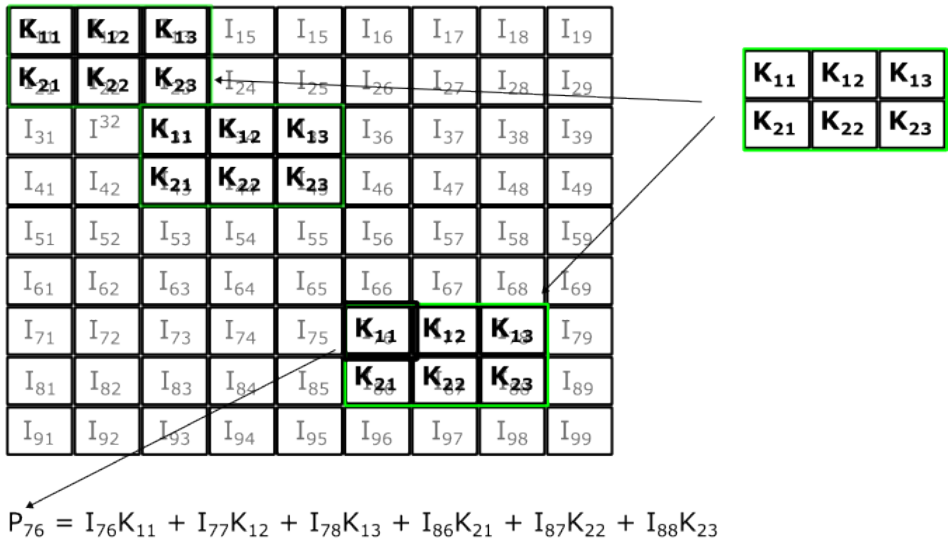


Figure 21: Principle of convolution operation on an image matrix, denoted I, with the kernel, denoted K. At each pixel P the convolution is computed, here the value of the pixel at P_{76}.

The result is a very well-defined image with sharp borders around the objects in the image, creating the desired result.

When tracking the particles, the pre-processing applied so far gave promising results in the particle tracking with Defocustracker.

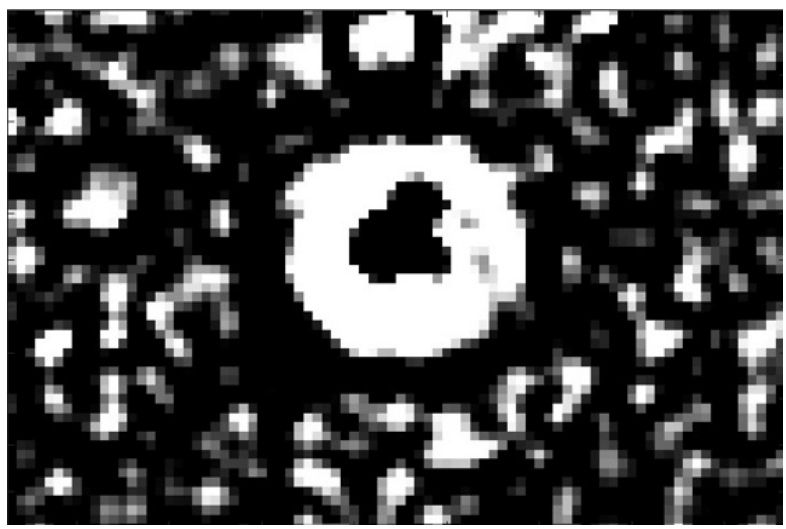


Figure 22: Image processed LoG filter. The result is a very detailed image.

Compared to before Laplacian of Gaussian filter was applied on the image, the image was now very well defined.

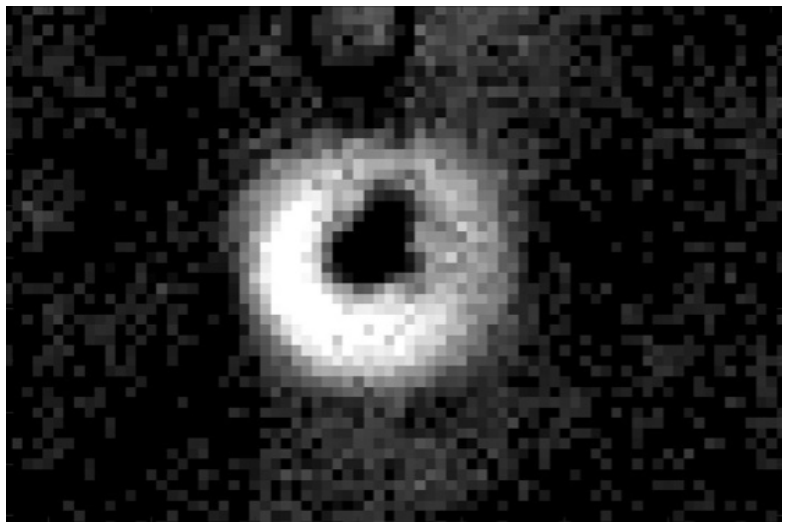


Figure 23: Image processed without LoG filter, only background subtraction applied.

Applying the LoG filter on the images, Defocustracker software was now able to track most particles correctly.

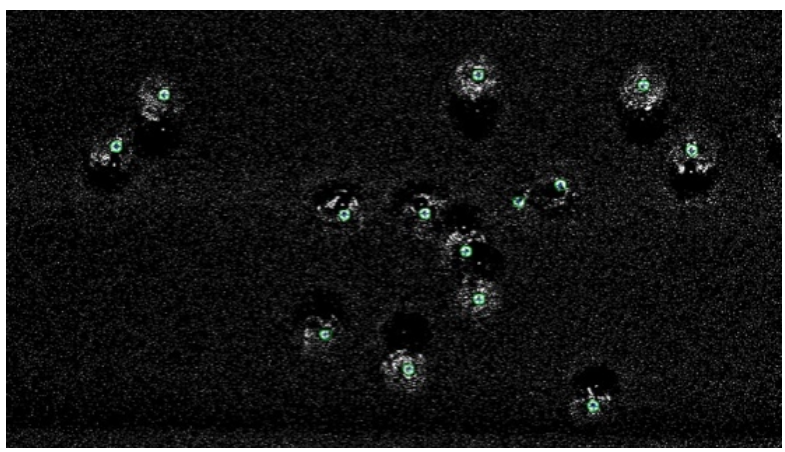


Figure 24: Defocustracker software detects most particles correctly after LoG filter was applied.

Before LoG was applied on the images, a lot of noise and shadows were registered as particles:

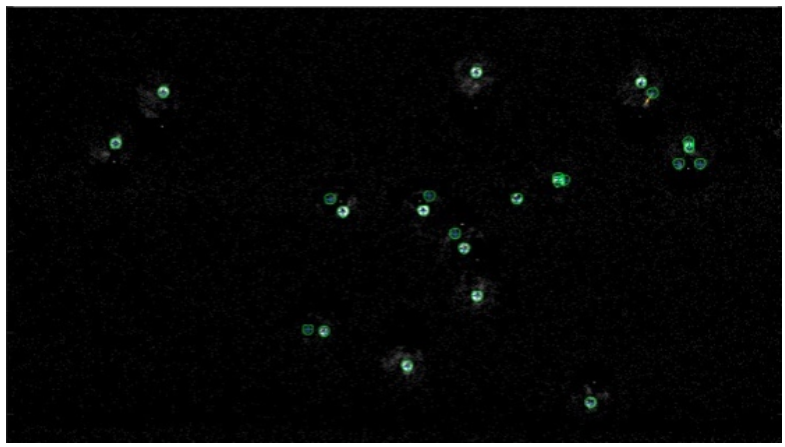


Figure 25: Defocustracker software registers noise and shadows as particles before LoG filter was applied.

4 Results

4.1 Assessment of pre-processing steps

An assessment of the performance of particle tracking after the individual pre-processing steps applied was made by selecting a set of 44 images. The first measure of quantification η_1, was the ratio between the number of shadows registered as particles and total number of particles:

$$\eta_1 = \frac{N_{shadows}}{N_{totalparticles} + N_{shadows}} \tag{5}$$

The three pre-processing steps were:

- background subtraction (S)
- imopen(I, strel) (I)
- LaPlacian of Gaussian(LoG)

They were combined in the following ways: background subtraction (S), and LoG filter (LoG), abbreviated SLoG. The second chain of processing was: background subtraction (S), imopen(I, strel) (I) and LoG (LoG), abbreviated SILoG. The second measure of quantification $\eta 2$ was the ratio of missed particles compared to the total amount of particles.

$$\eta_2 = \frac{N_{missed}}{N_{totalparticles}} \tag{6}$$

The quantification results after the different pre-processing steps can be seen in Table 1.

Applied processing	S	SI	SLoG	SILoG
η_1	31.90%	12,27%	0%	0%
η_2	0%	0%	0%	0%

Table 1: The ratios between the total amount of particles and the total amount of registrations, with different filters applied.

A clear a improvement can be seen when applying each pre-processing step to the images, as the average percentage of false particle detection drops from 32% to 12% when adding only the imopen(I, strel) to the background subtraction. When applying the LoG filter, there are drastic improvements. When LoG filter is applied, the number of shadows registered as particles drops to 0%.

The two most successful algorithms, the SLoG and SILoG algorithms, were examined more carefully. A larger set of images, 540, were inspected and all the shadows that were registered as particles were counted. This was the same measure of quantification used before, η_1, the ratio of shadows registered as particles and the actual number of particles.

There were only a few occurrences of this, and the results are seen in table 2.

Applied processing	SLoG	SILoG
η_1	0,08%	0,20%

Table 2: The ratios of shadows registered as particles and the total amount of particles.

The most reliable pre-processing algorithm was SLoG - background subtraction and Laplacian of Gaussian filtering.

With this pre-processing algorithm, further investigations were made to see how many of the true particles were not registered as particles in Defocustracker. The same subset of 540 images as before was inspected, and the missed particles counted in each image. The result can be seen in table 3

Applied processing	SLoG
η_2	2,17%

Table 3: The ratios of missed particles and the total amount of particles.

However, the camera at the lab setup was malfunctioning and had some pixel errors, resulting in empty pixel sections in the images. This resulted in deformations in some of the particles, and they could not be tracked, see figure 26. As mentioned earlier, the LoG process highlights regions of rapid intensity change.

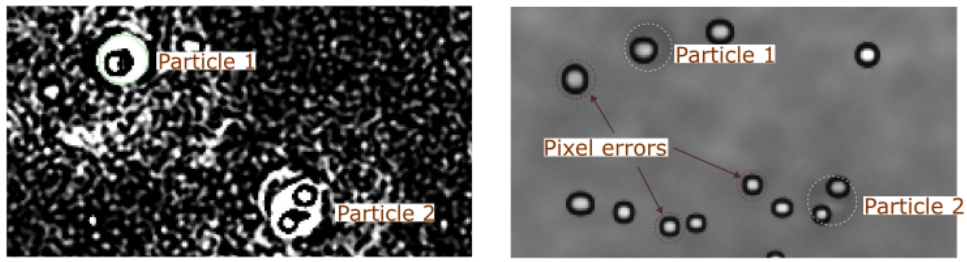


Figure 26: The same image: to the left it is processed with background subtraction and LoG, to the right is the original (here particle 1 and 2 are circled in grey dashed line). The camera pixel errors caused some deformations of the particles in the LoG processing, and these particles were missed.

There was one set of images available withouth the pixel errors, and when tracking the particles in those images, no particle was missed. The expected percentage of missed particles is thus zero, when no camera errors are present.

4.2 Comparing large sets of data to reference particles

From the data collected with Defocustracker - individual particle tracking data across many image sets - a sine curve was fitted using other Matlab scripts, described in Alexander Edthofer's masters thesis [7].

In Edthofers' scripts, the particle trajectories collected from Defocustracker are used to fit a sine curve. It shows how the particle velocities varies over the channel (the velocity being zero at the pressure node at the center of the channel). A reference set with fluorescent particles of known acoustic properties are used to determine the E_{ac} of the particular channel used in the experiment, comparing it to the fitted sine curve. The acoustic energy density, E_{ac}, were hence used as a measure of similarity between the reference data and our bright-field data. If the computed E_{ac} is the same for the data collected from dark-field and bright-field respectively, the particle tracking in bright-field would be considered accurate - we would expect the same E_{ac} for both data sets (the particles are dyed in both cases).
The results from fitting a sine to the particle trajectories and the mean squared error (MSE) can be seen in figure 27 and 28.

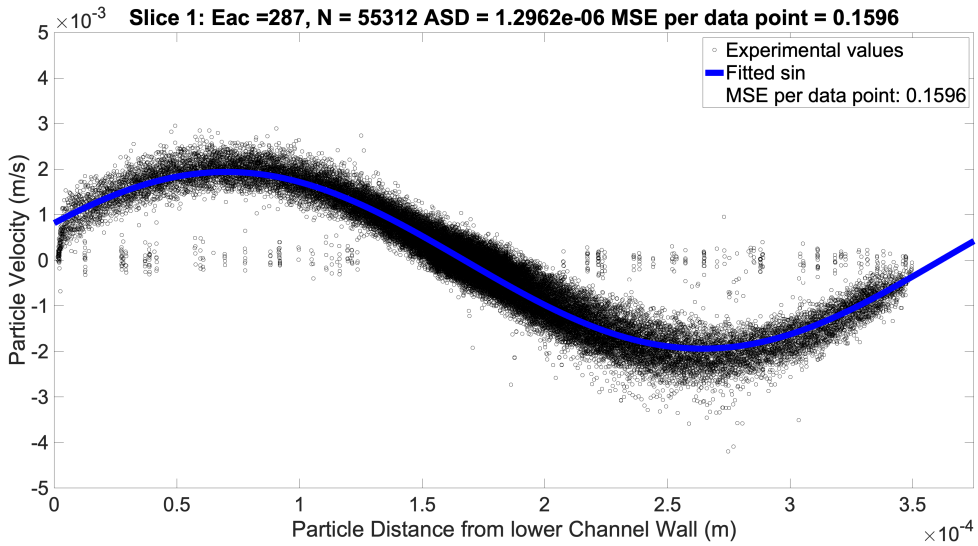


Figure 27: Reference particle (5 µm radius), dark-field dyed fluorescent, and the computed E_{ac}. N is the number of datapoints, and MSE is the average squared distance between the observed and predicted value.

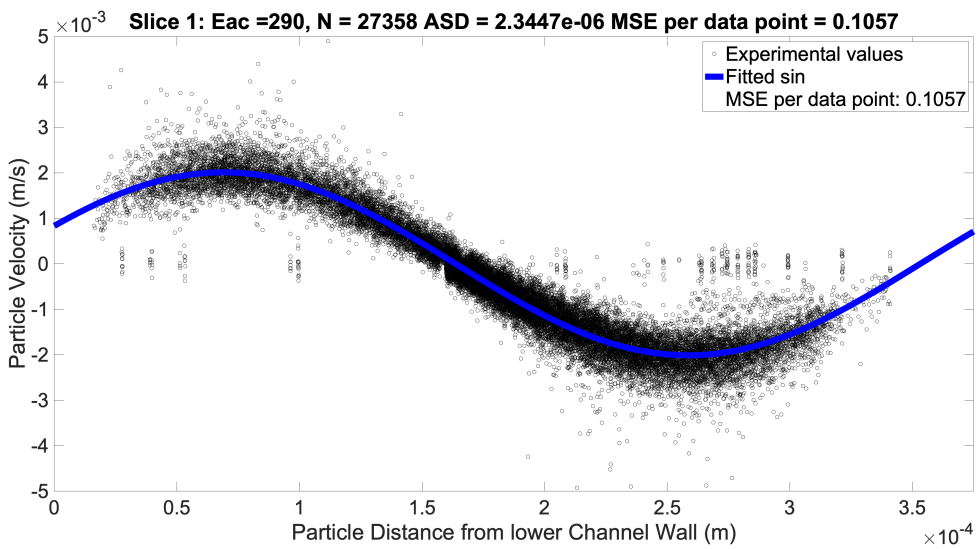


Figure 28: Bright-field particles (5 µm radius) processed with background subtraction and LoG filtering (SLoG) and the computed E_{ac}. N is the number of datapoints, and MSE is the average squared distance between the observed and predicted value.

The E_{ac} computed from the brigh-field data are not very far from the E_{ac} computed with the reference particle data. There is only a difference of 1,03%.

When using the same data as above, but filtering out the particles of low velocity, under $4 \cdot 10^{-4}$ m/s, we obtain the following fitted sine curves, figure 29 and 30:

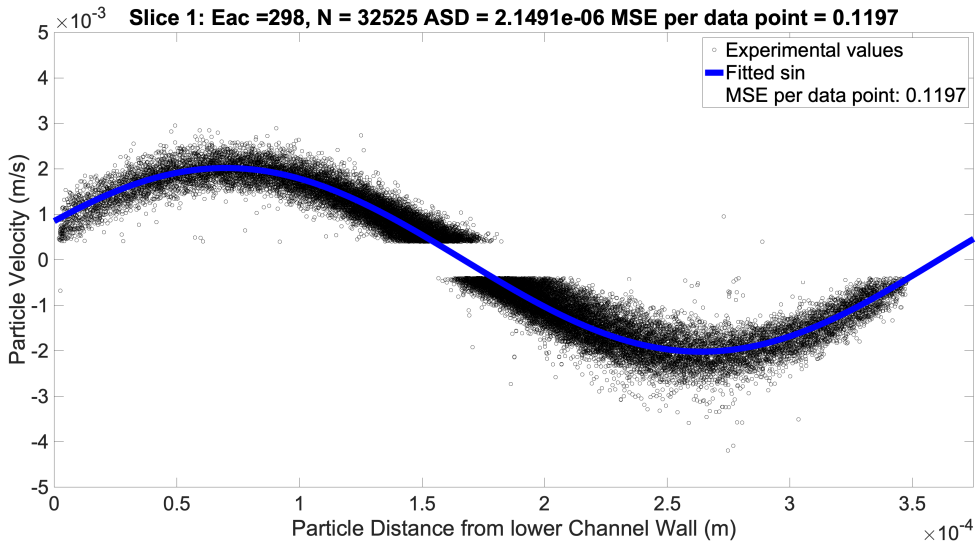


Figure 29: Reference particle (5 µm radius), dark-field dyed fluorescent, and the computed E_{ac}. N is the number of datapoints, and MSE is the average squared distance between the observed and predicted value. Particles of low velocity have been filtered out.

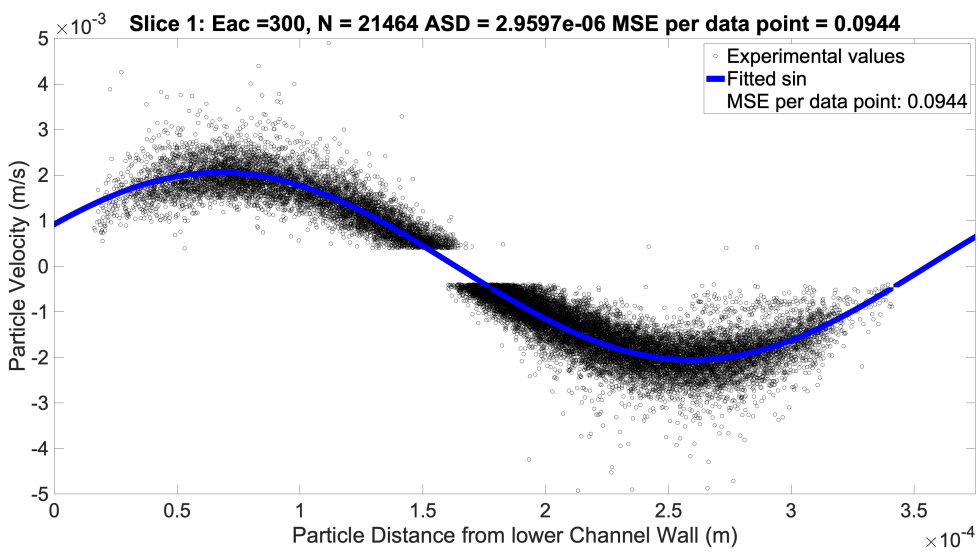


Figure 30: Bright-field particles (5 µm radius) processed with background subtraction and LoG filtering (SLoG) and the computed E_{ac}. N is the number of datapoints, and MSE is the average squared distance between the observed and predicted value. Particles of low velocity have been filtered out.

The difference in E_{ac} when low velocity particles are filtered out from the data points is 0,67%.

5 Conclusion

Four different image pre-processing steps were evaluated. Background subtraction with an image from the same set reduced the noise significantly, but noise from mainly the shadows of the particles were registered as particles in Defocustracker software. 32% of the registered signals were shadows. When adding an additional step, using the Matlab function imopen(I, strel)[10], first creating a filtered background image which is then subtracted from the background subtracted image, improves the tracking - the percentage of shadows registered as particles dropped to 12%.
Applying Laplacian of Gaussian filtering to the images improves the images drastically, and most particles are correctly tracked, and there are very few occurrences of shadow or noise registered as particles in Defocustracker. LoG was applied to background subtracted images (SLoG) and images processed with background subtraction and the imopen(I, strel) Matlab function.
Out of these preprocessing algorithms, the SLoG pre-processing had the best results - fewer shadows registered as particles.
The percentage of missed particles with the SLoG pre-processing was 2,17%, however this was mainly due to pixel errors in the camera used in the experiment. An image set without the pixel errors gave a result of 0% missed particles, which can be expected when processing images without these pixel errors.
When validating the collected tracking data over large data sets, we found that the E_{ac} computed with the dark-field tracking data was $288\frac{N}{m^2}$, and the corresponding E_{ac} for bright-field tracked with the SLoG algorithm, was $287\frac{N}{m^2}$. That is a difference of 1,03%, which is not significant.
When filtering out particles of low velocity, the difference in E_{ac} is 0,67%, which is also not significant.

References

[1] *Acoustofluidic separation of cells and particles* Wu et al. Microsystems & Nanoengineering 2019 5:32

[2] *Continuous separation of cells and particles in microfluidic systems* Lenshof& Laurell. Chem. Soc. Rev., 2010, 39, 1203-1217.

[3] *Two-hundredfold volume concentration of dilute cell and particle suspensions using chip integrated multistage acoustophoresis.* Maria Nordin, Thomas Laurell. Department of Biomedical Engineering, MultiPark: Multidisciplinary research focused on Parkinson´s disease. Year: 2012.

[4] *Using Acoustophoresis Cell Washing In The Immune Cell Radiolabeling Procedure* Adler et al. Journal of Nuclear Medicine June 2022, 63(supplement 2)2755.

[5] *Fluid Mechanics Vol. 6*, Landau L.D. & Lifschitz, Institute of physical problems, U.S.S.R Academy of sciences, 1987.

[6] *Acoustophoresis*, Lenshof, A., Laurell, T. (2015), Bhushan, B. (eds) Encyclopedia of Nanotechnology. Springer, Dordrecht.

[7] *Measuring Acoustic Mobility of Particles by Particle Tracking Velocimetry*, (2022) Alexander Edthofer, Masters' thesis in biomedical engineering

[8] *General defocusing particle tracking* R. Barnkob, C. J. Kähler, and M. Rossi. Lab on a Chip 15:17 (2015), pp. 3556–3560.

[9] *A fast and robust algorithm for general defocusing particle tracking*, Massimiliano Rossi & Rune Barnkob, Measurement Science and Technology, IOP Publishing, January 2020.

[10] *IMOPEN Morphologically open image*, Copyright 2018-2020 The Mathworks, Inc.

[11] *An improved edge detection using morphological Laplacian of Gaussian operator*, Anand Ashish et al, (2015) 2nd International Conference on Signal Processing and Integrated Networks (SPIN),